SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2817

SUSTAVI PREPORUČIVANJA TEMELJENI NA KORISNIČKIM SJEDNICAMA

Matej Luburić

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2817

SUSTAVI PREPORUČIVANJA TEMELJENI NA KORISNIČKIM SJEDNICAMA

Matej Luburić

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 11. ožujka 2022.

DIPLOMSKI ZADATAK br. 2817

Pristupnik:	**Matej Luburić (0036508139)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	izv. prof. dr. sc. Alan Jović

Zadatak: **Sustavi preporučivanja temeljeni na korisničkim sjednicama**

Opis zadatka:

Zbog sve većeg porasta kupovine preko interneta došlo je do povećane potrebe za pouzdanim preporučivanjima unutar kupovne sjednice korisnika, budući da namjere kupaca mogu biti različite ovisno o prilici, ali su one većinom slične unutar sjednice. Sustavi preporučivanja temeljeni na korisničkim sjednicama (engl. Session-based recommender systems) pružaju preporuke isključivo na temelju interakcija korisnika u trenutnoj sjednici i ne zahtijevaju postojanje korisničkih profila ili njihovih cjelokupnih povijesnih aktivnosti. Cilj takvih sustava je da na temelju niza kliknutih artikala u sjednici preporučuju artikle koji bi trebali biti zanimljivi korisniku. Preporuke se temelje na naučenim obrascima ponašanja korisnika na e-trgovini (engl. E-commerce). U ovom diplomskom radu opisat će se i primijeniti pristupe strojnog učenja i obrade teksta za koje je na temelju literature poznato da daju dobre rezultate na ovom problemu. Pristupi će se međusobno usporediti na slobodno dostupnom skupu podataka o korisničkim sjedinacama u e-trgovini.

Rok za predaju rada: 27. lipnja 2022.

SADRŽAJ

POPIS SLIKA

POPIS TABLICA

1. Uvod

Sustavi preporučivanja (engl. recommender systems) su aplikacije koje daju prilagođene prijedloge artikala korisnicima, obično s ciljem da im pomognu prevladati preopterećenost informacijama (artiklima) ili donijeti informirane odluke [10]. Oni su sve bitniji u suvremenom svijetu u e-trgovinama koje imaju velik broj artikala, internet oglašivačima raznih usluga, preporučivanju pjesmi korisnicima, i drugima, jer se s njima povećava angažman korisnika u vidu provedenog vremena na aplikaciji, rastu broja kupljenih proizvoda i samim time se povećava zadovoljstvo korisnika aplikacije, a poduzetnicima rastu prihodi.

Sustavi preporučivanja temeljeni na suradničkom filtriranju (engl. *collaborative filtering*) su temeljeni na pretpostavci da je poznata cijela povijest interakcija korisnika sa sustavom i ona se najčešće čuva u obliku matrice korisnik-artikal (engl. *user-item matrix*). Međutim, u mnogim aplikacijama informacije o povijesti interakcija korisnika sa sustavom nisu poznate. Neki od razloga su primjerice da korisnici nisu prijavljeni dok koriste aplikaciju, tek su registrirani korisnici ili su pak korisnici koji se još nisu željeli registrirati. U takvim slučajevima, mogu se iskoristiti tehnike koje iskorištavaju generalne obrasce ponašanja unutar zajednice korisnika [12]. Razlika je da umjesto dugoročnih preferencija korisnika, mogu se koristiti samo promatrane interakcije s korisnikom u trenutnoj sjednici za prilagodbu preporuka prema pretpostavljenim potrebama, preferencijama ili namjerama korisnika. Sustavi s tako postavljenim problemom se nazivaju sustavi preporučivanja temeljeni na korisničkim sjednicama (engl. *session-based recommender system*, dalje SBRS)[15].

Razmatranje interakcija unutar trenutne sjednice je također važno za aplikacije gdje prijavljeni korisnici često ponovno posjećuju e-trgovinu s kratkoročnom namjerom [13] gdje je zadaća sustava preporučivanja što bolje pretpostaviti namjeru korisnika ako ona postoji. Također, tradicionalni sustavi preporučivanja svakoj interakciji korisnika sa sustavom bi dali jednak značaj dok u stvarnosti imaju preferencije koje se mijenjaju s vremenom, te one najnovije interakcije su najrelevantnije.

Cilj sustava SBRS je kako kodirati, upamtiti općenite obrasce ponašanja zajednice

korisnika ovisno o kontekstu u kojem se nađu kako bi mogli predvidjeti sljedeće artikle za korisnike. Primjerice pretpostavka je da je vjerojatnije da će korisnik koji je kupio mobilni telefon također kupiti i zaštitni okvir za taj isti telefon prije nego zaštitni okvir za neki drugi mobilni telefon.

U ovom radu u poglavlju 2 formalno se definira zadatak za sustav SBRS, u poglavlju 3 uvode se osnovni modeli za primjenu u sustavu SBRS, a u poglavlju 4 objašnjavaju se modeli GRU4Rec i BERT4Rec koji su temeljeni na neuronskim mrežama. Opis skupova podataka je u poglavlju 5, a opis provedenih eksperimenata s rezultatima u poglavlju 6. Na kraju se kratko osvrće na potencijalne buduće smjerove razvoja sustava SBRS u poglavlju 7.

2. Definicija zadatka sustava preporučivanja temeljenih na korisničkim sjednicama

Zadatak sustava SBRS definira se na sljedeći način. Neka $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$ označava skup artikala, a lista $\mathcal{S}_u = [v_1^{(u)}, ..., v_t^{(u)}, ..., v_{n_u}^{(u)}]$ označava kronološki poredan niz interakcija anonimnog korisnika u, gdje je $v_t \in \mathcal{V}$ artikal s kojim je korisnik u bio u interakciji u vremenskom koraku t, a n_u je duljina korisničke sjednice za korisnika u. Za dani niz interakcije $\mathcal{S}_u$, sustav SBRS ima za cilj predvidjeti artikal koji će korisnik željeti vidjeti u vremenskom koraku $n_u + 1$. Navedeno se može formalizirati kao modeliranje vjerojatnosti svakog mogućeg artikla da će korisnik biti u interakciji s njim u vremenskom koraku $n_u + 1$ [18]:

$$p(v_{n_u+1}^{(u)} = v \mid \mathcal{S}_u) \tag{2.1}$$

Sustav preporučivanja predlaže listu artikala poredanih silazno od artikala s najvećom vjerojatnosti 2.1 prema onima s manjom vjerojatnosti. Duljina liste preporučenih artikala ovisi o problemu, a u ovom radu će se koristiti duljina liste od 20 artikala. Iz navedenog je vidljivo da je izrazito bitno mjesto artikala unutar liste koja se preporučuje i cilj sustava je da preporuči artikal s kojim će korisnik biti u interakciji što bliže početku (vrhu) liste.

Ideja ovakvog sustava preporučivanja je probati što preciznije odrediti namjeru korisnika u trenutnoj korisničkoj sjednici (engl. *intent prediction*) i na osnovu toga ponuditi korisniku relevantne artikle.

Kod sustava preporučivanja *session-aware* $\mathcal{S}_u$ predstavlja cijelu povijest interakcija korisnika sa sustavom, dok kod SBRS $\mathcal{S}_u$ predstavlja jednu korisničku sjednicu jednog anonimnog korisnika. Zbog toga je lista $\mathcal{S}_u$ manjeg kardinaliteta u SBRS u odnosu na sustave preporučivanja *session-aware*. Sustav SBRS ima teži problem za razliku od sustava preporučivanja *session-aware* jer uključivanjem informacija o preferencijama

korisnika može se povećati uspješnost preporuka. Uključivanje preferencija korisnika smanjiva prostor artikala iz kojeg se preporučuje zbog eliminiranja onih kategorija i artikala za koje je očito (iz povijesnih podataka) da ih korisnik ne voli ili mu nisu potrebni.

Kod primjene SBRS u e-trgovini više sjednica istog korisnika su omeđene neaktivnošću korisnika na određeni vremenski interval. Slijedi da, ako su dvije interakcije korisnika sa sustavom udaljene za više od unaprijed određenog vremeneskog intervala, interakcije će pripadati u različite sjednice.

Postoje različite vrste interakcija v: klik na artikl, dodaj u košaricu, dodaj na popis želja, itd. U ovom radu svi skupovi podataka će imati klikove na artikle kao jedinu vrstu interakcije između korisnika i artikala.

Postoje radovi u kojima je u zadatku preporučivanja unutar liste preporučenih sljedećih artikala postoji više relevantnih artikala za korisnika, a ne samo jedan. U radu [15] autori su uzeli da su relevantni artikli unutar liste preporučenih artikala oni na koje je korisnik kliknuo u nastavku sjednice. Na osnovu toga koristili su dodatne metrike koje vrednuju listu preporučenih artikala s više od jednog relevantnog artikla.

3. Osnovni modeli preporučivanja

Ovdje navedeni osnovni modeli temelje se na sličnosti k najbližih susjednih sjednica. Temeljeni su na jednostavnim pravilima s kojima se može jednostavno objasniti i razumjeti preporuke modela. Slijedi objašnjenje najefikasnijih osnovnih modela: SKNN, V-SKNN i STAN.

3.1. Model SKNN

Model temeljen na k najbližih susjednih korisničkih sjednica (engl. *session-based k nearest neighbors* – SKNN) uspoređuje trenutnu sjednicu s prošlim sjednicama kako bi preporučio artikle [11]. U prvom koraku model za trenutnu sjednicu s nalazi k najbližih sjednica N_s, a onda računa skor za artikal v sljedećim izrazom:

$$\text{skor}_{\text{SKNN}}(v, s) = \sum_{n \in N_s} sim(s, n) \times 1_n(v) \tag{3.1}$$

gdje je funkcija $sim(s_1, s_2)$ računa sličnost binarnih vektorskih reprezentacija dviju sjednica, a može biti sličnost kosinusa (formula 3.3), Jaccardov koeficijent sličnosti (formula 3.2), i dr. Funkcija $1_n(v) = 1$ ako susjedna sjednica n sadrži artikal v, a 0 inače.

Jaccardov koeficijent sličnosti se računa prema formuli:

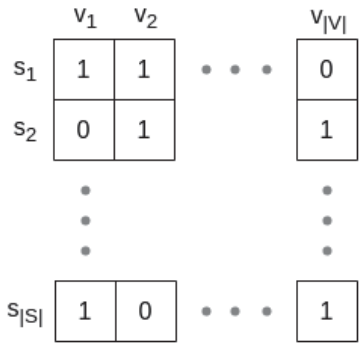
$$sim(s_1, s_2) = \frac{|s_1 \cap s_2|}{|s_1 \cup s_2|}, \tag{3.2}$$

gdje $|s_1 \cap s_2|$ je broj zajedničkih artikala između sjednica s_1 i s_2, a $|s_1 \cup s_2|$ broj jedinstvenih artikala u obje sjednice. Dakle, na Jaccardov koeficijent sličnosti ne utječe ponavljanje artikala unutar sjednica. Sličnost kosinusa se računa kao:

$$sim(s_1, s_2) = \frac{\boldsymbol{s}_1 \boldsymbol{s}_2}{\sqrt{l(s_1) l(s_2)}}, \tag{3.3}$$

gdje $\boldsymbol{s}_i$ predstavlja vektorsku reprezentaciju za artikal i i to je redak iz matrice sjednica-artikal (slika 3.1). Član $l(s_i)$ predstavlja duljinu sjednice s_i . Za razliku od Jaccardovog

koeficijenta sličnosti, sličnost kosinusa uzima u obzir duljinu sjednice, a ne broj jedinstvenih artikala unutar sjednica, pa na sličnost kosinusa utječe ponavljanje artikala unutar sjednica (povećava nazivnik iz formule 3.3). U radu [11] testirali su više funkcija sličnosti i dobili su da je najbolja sličnost kosinusa s binarnim variablama. Traženje



Slika 3.1: Matrica sjednica-artikal

najbližih sjednica N_s među svim sjednicama u skupu podataka može biti dugotrajno i samim time neprimjenjivo u primjeni gdje korisnici ne smiju čekati preporuke. Autori u [11] su napravili obrnuti indeks u kojem se svakom artiklu pridružuje lista sa svim identifikatorima sjednica u kojem se ti artikli nalaze. Zatim se uzimaju sve sjednice u kojima je barem jedan artikal iz trenutne sjednice s koje onda čine m potencijalnih susjednih sjednica, a k sjednica se odabire heuristikom. Za sustav SBRS u e-trgovini autori su zaključili da je najbolja heuristika izabrati k najnovijih sjednica koja prepoznaje najnovije trendove među korisnicima. Zatim se iz k najbližih sjednica dobiva skup jedinstvenih artikala R. Za svaki artikal $v \in R$ se računa $\mathrm{skor}_{\mathrm{SKNN}}(v, s)$ na osnovu kojeg se preporučuje lista sljedećih artikala.

Veći skor će imati oni artikli koji se češće nađu u sudjednoj sjednici s trenutnim artiklom v (više puta se pribraja $sim(s, n)$) ili oni artikli koji se nalaze u susjednim sjednicama koje su najsličnije trenutnoj (veći iznos $sim(s, n)$).

Model SKNN je praktički metoda suradnčkog filtriranja (engl. *collaborative filtering*) gdje se umjesto matrice korisnik-artikal koristi matrica sjednica-artikal s binarnim varijablama (slika 3.1).

3.2. Model V-SKNN

Redoslijed artikala unutar trenutne sjednice može biti važan, jer osobito pri dugim sjednicama, korisnik može promjeniti namjeru i onda artikli s početka sjednice počinju biti manje relavantni ili irelevantni [14]. Skor aktikla v na temelju trenutne sjednice s modela V-SKNN se računa kao [1]:

$$\begin{gathered}\text{skor}_{\text{V-SKNN}}(v,s) = \sum_{n \in N_s} (sim(s,n) + z) \cdot w(v,s) \cdot 1_n(v), \\ z = sim(s,n) \cdot idf(v) \cdot IDF\_Weighting,\end{gathered} \tag{3.4}$$

gdje je funkcija $w(v,s)$ koja regulira važnost pozicije artikla v unutar trenutne sjednice s. Vrijednosti funkcije $w(v,s)$ će padati prema početku trenutne sjednice, a brzina padanja vrijednosti može se regulirati različitim funkcijama (eksponencijalna, kvadratna, linearna, itd.).

Funkcija $idf(v)$ predstavlja inverznu frekvenciju artikla (engl. *inverse document frequency – IDF*) u sjednicama, a računa se kao:

$$idf(v) = \log\left(\frac{|\mathcal{S}|}{\sum_{i=1}^{|\mathcal{S}|} 1_{s_i}(v)}\right), \tag{3.5}$$

gdje je brojnik logaritma ukupan broj sjednica, a nazivnik broj sjednica u kojima se nalazi artikal v. Slijedi da artikli koji se nalaze u manjem udjelu sjednica imaju veću vrijednost funkcije $idf(v)$, što znači da će u konačnici imati veći skor od artikala koji se češće pojavljuju u sjednicama. S parametrom $IDF\_Weighting$ se regulira hoće li se uopće koristiti ponderiranje ($IDF\_Weighting = 0$) i jačina ponderiranja ($IDF\_Weighting > 0$).

Na osnovu formule 3.4 slijedi da na skor artikla iz susjedne sjednice utječe: nedavnost artikla iz trenutne sjednice s kojim se nalazi u susjednoj sjednici (vrijednost funkcije $w(v,s)$), broj pojavljivanja artikla u sjednicama iz skupa za učenje (parametar z), broj zajedničkih artikala susjedne i trenutne sjednice (vrijednost sličnosti $sim(s,n)$) te broj pojavljivanja artikla v u skupu susjednih sjednica N_s (koliko puta funkcija $1_n(v) = 1$; broj pribrojnika sume).

[1] Zaključeno na osnovu kôda `https://github.com/rn5l/session-rec`

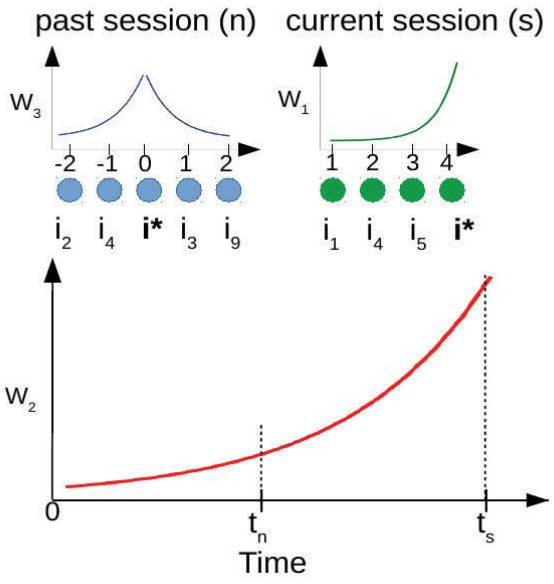
3.3. Model STAN

Model STAN (engl. *sequence and time aware neighborhood*) uzima u obzir:

1. poziciju artikla unutar trenutne sjednice
2. recentnost prošle sjednice u odnosu na trenutnu i
3. poziciju artikla u odnosu na artikal iz trenutne sjednice u susjednoj sjednici [4].

Tako, model STAN ima sljedeća tri parametra koji modeliraju sljedeća tri svojstva :

1. zaboravljenje artikala unutar trenutne sjednice,
2. zaboravljanje prošlih sjednica i artikala unutar njih i
3. zaboravljanje artikla koji su u odnosu na artikal iz trenutne sjednice udaljeniji u prošloj sjednici [4].



Slika 3.2: Parametri modela STAN [4]

Parametar w_1 (slika 3.2). Artikli bliži kraju trenutne sjednice su važniji za sljedeću preporuku. Svakom artiklu i iz trenutne sjednice s se pridružuje:

$$w_1(i,s) = \exp\left(\frac{p(i,s) - l(s)}{\lambda_1}\right) \tag{3.6}$$

gdje je $p(i,s)$ pozicija artikla i unutar trenutne sjednice $l(s)$ je duljina trenutne sjednice, a $\lambda_1 > 0$. Zatim se računa sličnost kosinusa trenutne sjednice sa susjednim sjednicama s_j po formuli :

$$sim_1(s, s_j) = \frac{\boldsymbol{s}_w \boldsymbol{s}_j}{\sqrt{l(s)l(s_j)}} \tag{3.7}$$

gdje je $\boldsymbol{s}_w$ podešeni vektor trenutne sjednice, $\boldsymbol{s}_j$ binarni vektor susjedne sjednice.

Parametar w_2 (slika 3.2) određuje koliko su sjednice vremenski bliže trenutnoj sjednici važnije. Parametar w_2 se računa:

$$w_2(s_j \mid s) = \exp\left(-\frac{t(s) - t(s_j)}{\lambda_2}\right) \tag{3.8}$$

gdje $t(s) - t(s_j)$ predstavlja vremenski razmak sjednica, $\lambda_2 > 0$. Skup susjednih sjednica $\mathcal{N}(s)$ čine sjednice koje imaju najveću sličnost sim_2:

$$sim_2(s, s_j) = sim_1(s, s_j) w_2(s_j \mid s). \tag{3.9}$$

Parametar w_3 (slika 3.2). Sljedeći artikli preporučuju se isključivo iz skupa $\mathcal{N}_s$ s tim da se udaljenost tih aritikala s artiklima iz trenutne sjednice unutar susjedne sjednice uzima u obzir na sljedeći način:

$$w_3(i \mid s, n) = \exp\left(-\frac{|p(i,n) - p(i^*,n)|}{\lambda_3}\right) I_n(i), \tag{3.10}$$

gdje je i^* artikal koji se nalazi u trenutnoj i susjednoj sjednici, $\lambda_3 > 0$, a I_n je indikatorska funkcija koja iznosi 1 ako je artikal i u susjednoj sjednici n, a 0 inače. Relevantnost artikla i iz susjedne sjednice n je:

$$rel(i \mid s, n) = sim_2(s, n) w_3(i \mid s, n) \tag{3.11}$$

dok je njegova ukupna relevantnost uz sve susjedne sjednice:

$$skor_{STAN}(i, s) = \sum_{n \in \mathcal{N}(s)} rel(i \mid s, n) \tag{3.12}$$

Parametri λ_i određuju brzinu zaboravljanja artikala (sjednica). Tako, parametrima $\lambda_i \ll 1$ daje se veća važnost nedavnijim artiklima (sjednicama), dok s $\lambda_i \to \infty$ svi artikli (sjednice) su jednako važne i nema zaboravljanja, a tako se dobiva model SKNN. Svi navedeni parametri λ_i mogu biti podesivi ovisno o području primjene.

U radu [4] je pokazano da redoslijed artikala (prvi parametar) najviše utječe na rezultate i da je model STAN bolji od modela SKNN na kraćim (< 5 artikala) i duljim sjednicama (≥ 5), a razlika se povećava na duljim sjednicama gdje se pokazuje da je bitna pozornost na nedavnije artikle unutar trenutne sjednice.

4. Modeli preporučivanja

Bitna prednost neuronskih modela nad osnovnim modelima je to što je lako uključiti metapodatke artikala u ugradbene vektore artikala koji se onda dalje mogu iskoristiti za lakše preporučivanje sljedećih artiakala, dok kod osnovnih modela to nije moguće ili je teško modelirati. Metapodaci mogu biti kategorički (npr. boja artikla), brojčani (npr. cijena), tekstualni (opis artikla), fotografije, i dr. U nastavku slijedi objašnjenje modela: GRU4Rec i BERT4Rec.

4.1. Model GRU4Rec

Model GRU4Rec koristi propusnu povratnu ćeliju (engl. *gated recurrent unit* – GRU). Ćelija GRU podvrsta je rekurentnih povratnih neuronskih mreža (engl. *recurrent neural networks – RNN*) koje se koriste za obradu sekvencijskih podataka. Ćelija GRU, u odnosu na jednostvanu RNN, rješava problem iščezavajućeg i eksplodirajućeg gradijenta [2]. Vektor skrivenih značajki $\boldsymbol{h}_t$ računa se kao:

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t)\boldsymbol{h}_{t-1} + \boldsymbol{z}_t\hat{\boldsymbol{h}_t} \tag{4.1}$$

gdje je propusnica ažuriranja (engl. *update gate*) $\boldsymbol{z}_t$ jednaka:

$$\boldsymbol{z}_t = \sigma(W_z\boldsymbol{x}_t + U_z\boldsymbol{h}_{t-1}), \tag{4.2}$$

a prijedlog budućeg stanja $\hat{\boldsymbol{h}_t}$ je:

$$\hat{\boldsymbol{h}_t} = \tanh(W\boldsymbol{x}_t + U(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})), \tag{4.3}$$

gdje je propusnica resetiranja (engl. *reset gate*) $\boldsymbol{r}_t$ jednaka:

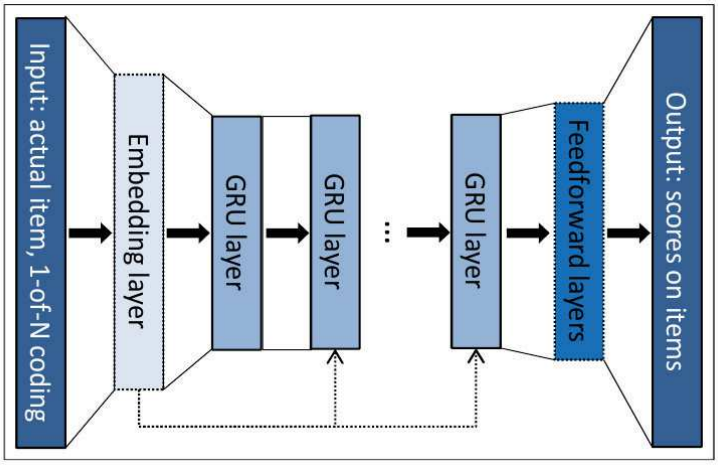
$$\boldsymbol{r}_t = \sigma(W_r\boldsymbol{x}_t + U_r\boldsymbol{h}_{t-1}) \tag{4.4}$$

Ćelija GRU tako preko propusnica za svaki ulazni podatak regulira koje informacije će zadržati, koje dodati (propusnica ažuriranja), a koje zaboraviti (preopusnica resetiranja).

Kako su korisničke sjednice sekvencijski podaci, u radu [9] se uvodi primjena ćelije GRU u sustave preporučivanja. Opća arhitektura GRU4Rec je na slici 4.1.

U arhitekturi modela GRU4Rec postoje 3 načina modeliranja ulaza u GRU sloj: bez ugradbene matrice, s ugradbenom matricom i kad ugradbenu matricu čine težine izlaznog sloja. Ako na ulazu u GRU sloj ne postoji ugradbena matrica, za artikle se koristi vektor indikatorski varijabli (engl. *one-hot encoding*) koji se onda s potpuno povezanim slojem sažima na veličinu vektora skrivenih značajki h. Ako se koristi ugradbena matrica ona se indeksira identifikatorom ulaznog artikla što daje ugradbeni vektor tog artikla koji se dalje šalje na ulaz GRU sloja. Kad ugradbenu matricu čine težine izlaznog sloja ugradbeni vektori artikala se uče za vrijeme učenja modela. Neka je pretpostavka da su dimenzije ugradbenih vektora jednake dimenziji vektora skrivenih značajki ćelije GRU i iznose d, a ukupan broj artikala neka je $|V|$. Usko grlo memorije je matrica W_x (dimenzija $d \times |V|$) u prvom slučaju, ugradbena matrica E (dimenzije $d \times |V|$) u drugom slučaju i u sva tri slučaja izlazna matrica W_y (dimenzije $d \times |V|$) Slijedi, ako se koriste težine izlaznog sloja kao ugradbena matrica koristi se značajno manje memorije [7].

Nakon ulaza slijedi jedan ili više slojeva s ćelijama GRU. Ako ima više ćelija GRU u nizu, izlaz iz jedne sloja je ulaz u sljedeću ćeliju GRU. Također je moguće imati rezidualne veze od ugradbenog sloja do ćelija GRU, a u praksi se pokazalo da daje bolje rezultate [9]. Nakon toga slijedi potpuno povezani sloj (engl. *fully connected feedforward layer*) koji transformira vektor skrivenih značajki h zadnjeg sloja u prostor svih artikala.



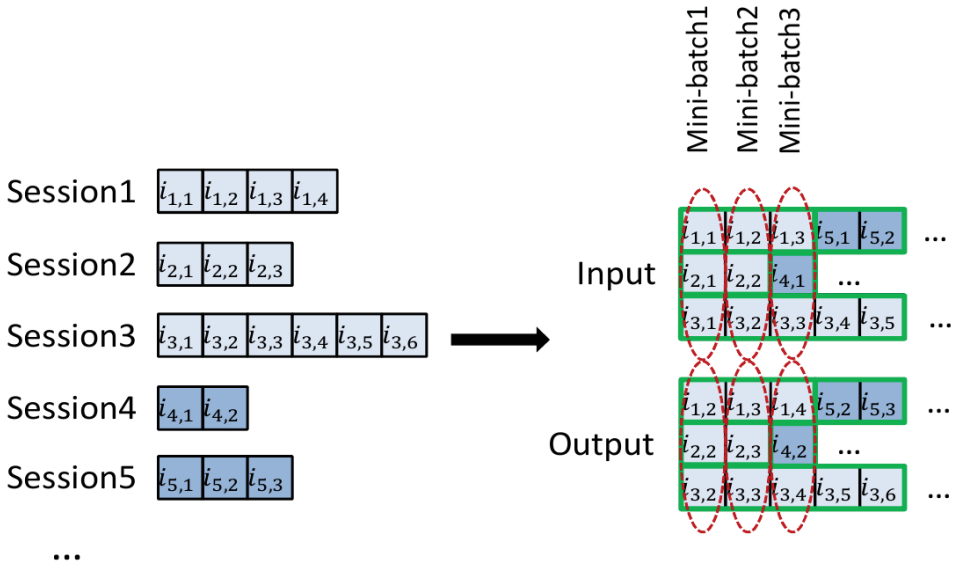
Slika 4.1: Općenita arhitektura modela GRU4Rec [9]

.

4.1.1. Formiranje mini grupa za učenje

Na ulaznom sloju vektor indikatorskih varijabli se preslikava ($1 \times |V|$) u vektor značajki skrivenog sloja, a na izlaznom sloju skrivene značajke se transformiraju u prostor svih artikala kako bi se dobila izglednost za svaki artikal. Problem postaje veliki broj mogućih artikala koji može biti nekoliko stotina tisuća (milijuna) i zbog toga preslikavanja na ulazu i na izlazu postaju usko grlo algoritma. Rješenje ovog problema prvi put je uveden u [16] gdje autori koriste uzorkovanje negativnih primjera (svi koji nisu ciljni artikal) i time se ažurira samo dio težina u izlaznom i ulaznom sloju.

Autori u [9] koriste pametan način raspodijele artikala iz sjednica u mini grupe (Slika 4.2) čime se postiže paralelizacija. Sjednice se prvo sortiraju po vremenu nas-



Slika 4.2: Formiranje mini grupa za učenje [9]

tanka. Zatim prvi artikli iz prvih X sjednica čine prvu mini grupu. Ako neka sjednica završi na njen kraj se dodaje nova sjednica, zatim se, uz pretpostavku nezavisnosti sjednica, resetira se skriveni sloj. Negativni primjeri uzorkuju se iz ciljnih artikala mini grupe. Primjerice sa slike 4.2 za artikal $i_{1,1}$ iz mini grupe 1, ciljni artikal je $i_{1,2}$ a negativni primjeri su $i_{2,2}$ i $i_{3,2}$. Pokazuje se da je ovakvo negativno uzorkovanje zapravo temeljeno na popularnosti jer izglednost artikala da budu sljedeći artikli trenutnoj mini grupi proporcionalna je popularnosti tih artikala [9].

Svojstvo funkcija gubitka pri rangiranju da uče jedino kad skor ciljnog artikla ne dominira nad ostalim negativnim primjerima, pa je tako izrazito važno da negativni primjeri imaju što veći skor. Popularni artikli često imaju veliki skor, pa je dobra strategija uzorkovanje negativnih primjera temeljeno na popularnosti. Problem koji nastaje

kad algoritam nauči davati puno veći skor ciljnom artiklu u odnosu na popularne dok zanemaruje nepopularne artikle s velikim skorom. Rješenje je uzorkovanje dodatnih negativnih primjera s parametrom α kojim se regulira vrsta uzorkovanja. S $\alpha = 0$ dobiva se uniformno uzorkovanje dok s $\alpha = 1$ uzorkovanje temeljeno na popularnosti [7].

4.1.2. Funkcije gubitka pri rangiranju

Kod svakog neuronskog modela funkcija gubitka je izrazito bitna jer ona određuje kvalitetu gradijenata, a koji se onda koriste za učenje modela. Stabilni gradijenti funckije gubitka posebno su bitni kod rekurentnih neuronskih modela zbog duge propagacije gradijenata unatrag kroz vrijeme.

U radu [9] se uvodi nova funkcije gubitka po parovima (engl. *pairwise loss function*) TOP1. Funkcija gubitka po parovima uzima parove skorova između pozitivnog i negativnih primjera te potiče da pozitivni primjer ima bolji rang od negativnih. Funkcija gubitka **TOP1** se računa kao:

$$L_s = \frac{1}{N_S} \sum_{j=1}^{N_S} \sigma(r_{s,j} - r_{s,i}) + \sigma(r_{s,j}^2), \tag{4.5}$$

gdje je $r_{s,j}$ skor negativnog primjera iz skupa negativnih primjera N_S, a $r_{s,i}$ je skor pozitivnog primjera. Skor predstavlja izglednost artikla da je on sljedeći artikalu sjednici. Član $\sigma(r_{s,j}^2)$ je regularizacijski faktor čiji je cilj pritezanje na 0 skora za negativne primjere. Iz formule (4.5) je vidljivo da će veći gubitak biti kada je skor negativnog primjera veći od skora pozitivnog primjera, a što je skor pozitivnog primjera $r_{s,i}$ bliže 1, gubitak teži u 0.

Nedostatak funkcije gubitka TOP1 što se povećanjem uzorka negativnih primjera povećava vjerojatnost odabira artikala s malim skorom koji ne doprinose učenju dok se ukupni gubitak dijeli brojem negativnih primjera. Autori u [7] uvode novu funkciju gubitka **TOP1-MAX**:

$$L_{TOP1-MAX} = \sum_{j=1}^{N_S} s_j \left(\sigma(r_j - r_i) + \sigma(r_j^2) \right), \tag{4.6}$$

gdje je s_j iznos funkcije softmax negativnog artikla među negativnim artiklima. Slijedi, ako je skor negativnog artikla r_j mnogo manji od skora najvećeg negativnog artikla, s_j će biti približno 0, što znači da će onda negativni primjeri s većim skorom imati veću težinu, jer je $\sum s_j = 1$.

Autori u [7] navode da je novi način uzorkovanja negativnih primjera i nova funkcija gubitka donose rast mjere HR@20 od 21% u odnosu na osnovni GRU4Rec ([9]) i što ga čini najboljim modelom GRU4Rec u sustavima preporučivanja temeljenim na korisničkim sjednicama.

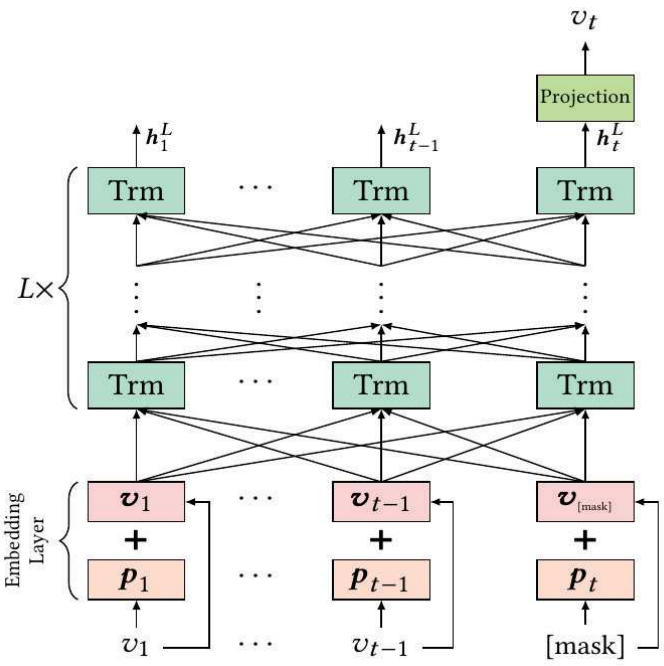
4.2. Model BERT4Rec

BERT4Rec (engl. *Bidirectional Encoder Representations from Transformer for Recommendation*) je model koji je nastao na osnovu modela BERT koji dolazi iz područja obrade prirodnog jezika (engl. *natural language processing – NLP*). U odnosu na osnovni BERT koji je prednaučen na velikoj količini podataka jer postoji dijeljeno znanje o jeziku između različitih skupova podataka, to nije moguće u sustavima preporučivanja gdje je svaki skup podataka ima svoje pravilnosti. Također, kako se BERT4Rec primjenjuje na samo jednoj sjednici (jednoj rečenici) nema potrebe za ugradbenim vektorom segmenta i gubitka predviđanja sljedeće rečenice (engl. *next sentence loss*) [18]. Model BERT4Rec u radu [18] je izvorno primjenjen u sustavu preporučivanja *session-aware* gdje je potrebno predvidjeti sljedeći artikal na koji će korisnik kliknuti uz poznavanje njegove čitave povijesti klikova. To je prvi rad koji je uveo duboki dvosmjerni sekvencijski model u sustave preporučivanja. Pretpostavka modela RNN (GRU4Rec je jedan od njih) je da su artikli strogo poredani, pa stoga za učenje i testiranje koriste jedino smjer s lijeva k desno unutar sjednice. Uvođenje dvosmjernog modela kod kojeg svaki artikal utječe na drugi unutar sjednice neovisno o udaljenosti povećava kapacitet modela da "shvati" kontekst korisničke sjednice.

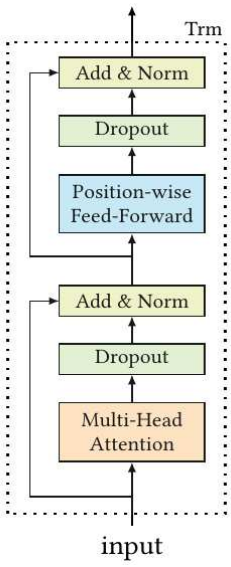
4.2.1. Arhitektura modela BERT4Rec

Arhitekturu modela BERT4Rec možemo podijeliti na 3 sloja: transformerski sloj (tamno zeleni pravokutnici na slici 4.3), ugradbeni (engl. *embedding*) sloj (crveni i bež pravokutnici sa slike 4.3) i izlazni sloj (svijetlo zeleni pravokutnik sa slike 4.3) .

Transformerski sloj na ulaz dobiva ulaznu sekvencu duljine t, a onda iterativno i paralelno računa latentne značajke $\boldsymbol{h_i^l} \in \mathbb{R}^d$ za poziciju i i transformerski sloj l. latentne značajke svih pozicija se spajaju u matricu $\boldsymbol{H^l} \in \mathbb{R}^{t \times d}$ radi paralelnog izračunavanja. Transformerski sloj (na slici 4.4) TRM sastoji se od dva podsloja: višestruka samopozornost (engl. *multi-head self attention*, dalje MH) i pozicijske unaprijedne neuronske mreže (engl. *position-wise feed-forward*, dalje PFFN) (slika 4.4). Podsloj MH prvo projicira matricu $\boldsymbol{H^l}$ u h podprostora onda paralelno nad svakim od njih primjenjuje



Slika 4.3: Arhitektura modela BERT4Rec [18]



Slika 4.4: Jedan transformerski blok [18]

funkciju pozornosti, zatim konkatenira rezultate koje na kraju ponovno projiciraju u prostor $t \times d$:

$$\begin{aligned} \text{MH}\left(\boldsymbol{H}^l\right) &= [\text{glava}_1; \text{glava}_2, \ldots, \text{glava}_h]\boldsymbol{W}^O \\ \text{glava}_i &= \text{Pozornost}\left(\boldsymbol{H}^l\boldsymbol{W}_i^Q, \boldsymbol{H}^l\boldsymbol{W}_i^K, \boldsymbol{H}^l\boldsymbol{W}_i^V\right) \end{aligned} \tag{4.7}$$

gdje projekcijske matrice $\boldsymbol{W}_i^Q \in \mathbb{R}^{d\times\frac{d}{h}}$, $\boldsymbol{W}_i^K \in \mathbb{R}^{d\times\frac{d}{h}}$, $\boldsymbol{W}_i^V \in \mathbb{R}^{d\times\frac{d}{h}}$ sadrže težine koje se uče posebno za svaki sloj l. Funkcija Pozornost je skalirani skalarni produkt:

$$\text{Pozornost}\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d/h}}\right)\boldsymbol{V} \tag{4.8}$$

gdje su $\boldsymbol{Q}$ upiti, $\boldsymbol{K}$ ključevi, a $\boldsymbol{V}$ vrijednosti koji su nastali linearnom projekcijom matrice $\boldsymbol{H^l}$. Izraz $\sqrt{d/h}$ je tzv. temperatura koja služi da bi se dobila glađa distribucija funkcije pozornosti, a time izbjegli mali gradijenti pri optimizaciji parametara.
Podsloj PFFN (slika 4.4) primjenjuje dodatne transformacije uz uključivanje nelinearnosti funkcijom linearne jedinice Gaussove pogreške (engl. *Gaussian Error Linear Unit* – GELU). Funkcije podsloja su sljedeće:

$$\begin{aligned} \text{PFFN}\left(\boldsymbol{H^l}\right) &= [\text{FFN}\left(\boldsymbol{h}_1^l\right)^T; ...; \text{FFN}(\boldsymbol{h}_t^l)^T]^T \\ \text{FFN}\left(\boldsymbol{x}\right) &= \text{GELU}\left(\boldsymbol{x}\boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)}\right)\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)} \end{aligned} \tag{4.9}$$

gdje su $\boldsymbol{W}^{(1)} \in \mathbb{R}^{d\times 4d}$, $\boldsymbol{W}^{(2)} \in \mathbb{R}^{4d\times d}$, $\boldsymbol{b}^{(1)} \in \mathbb{R}^{4d}$, $\boldsymbol{b}^{(2)} \in \mathbb{R}^{d}$, parametri koji su dijeljeni između svih pozicija t.
Konačni izlazi svakog podsloja dobivamo s:

$$\text{LN}\left(\boldsymbol{x} + \text{Dropout}\left(\text{podsloj}\left(\boldsymbol{x}\right)\right)\right) \tag{4.10}$$

gdje na izlaze svakog podsloja prvo se primjenjuje regularizacija metodom isključivanja čvorova grafa (engl. *dropout*), na to se dodaju ulazi podsloja i na kraju se provodi normalizacija međurezultata LN.
Sažeto, transformerski sloj se može prikazati sljedećim funkcijama [18]:

$$\begin{aligned} \boldsymbol{H}^l &= \text{Trm}\left(\boldsymbol{H}^{l-1}\right), \forall i \in [1, ..., L] \\ \text{Trm}\left(\boldsymbol{H}^{l-1}\right) &= \text{LN}\left(\boldsymbol{A}^{l-1} + \text{Dropout}\left(\text{PFFN}\left(\boldsymbol{A}^{l-1}\right)\right)\right) \\ \boldsymbol{A}^{l-1} &= \text{LN}\left(\boldsymbol{H}^{l-1} + \text{Dropout}\left(\text{MH}\left(\boldsymbol{H}^{l-1}\right)\right)\right) \end{aligned} \tag{4.11}$$

Kako bi model bio svjestan redoslijeda artikala s ulaza, **ugradbeni sloj** ugrađuje informaciju o poziciji artikala na sljedeći način:

$$\boldsymbol{h}_i^0 = \boldsymbol{v}_i + \boldsymbol{p}_i \tag{4.12}$$

gdje je $\boldsymbol{v}_i \in \boldsymbol{E}$ ugradbeni vektor artikla, a $\boldsymbol{p}_i \in \boldsymbol{P}$ je pozicijski ugradbeni vektor. Ugradbeni vektor artikla u ovom radu je popunjen slučajnim brojevima iz određenog intervala, dok u naprednijoj verziji modela u njemu mogu biti pohranjene sažete informacije o artiklu, kao što su: slika, cijena, opis, komentari o proizvodu i dr. Pozicijski ugradbeni vektori $\boldsymbol{P} \in \mathbb{R}^{N \times d}$ se također uče, a veličina matrice ovisi o maksimalnom broju artikala N koje model može učitati odjednom. Ako su korisničke sjednice duže od N artikala uzima se zadnjih $N-1$ artikala s još posebnim tokenom $[\text{mask}]$ koji predstavlja artikal kojeg model treba pogoditi.

Na kraju, nakon L slojeva transformera dobiva se matrica latentnih značajki H^L. Matrica latentnih značajki H^L sadrži značajke koje model smatra bitnim. Transformerki sloj L uzima latentne značajke iz prethodnog $L-1$ sloja i nad njima ponovno traži određene pravilnosti. Zbog toga transformerski sloj na razini L ima veći nivo apstrakcije od $L-1$ sloja. Ako model treba pogoditi koji artikal se nalazi na poziciji t, uzima se vektor h_t^L, koji je ulazni vektor **izlaznog sloja**, koji se zatim projicira da bi se dobila vjerojatnost za svaki artikal iz skupa svih artikala $\mathcal{V}$:

$$P\left(v\right) = \text{softmax}\left(\text{GELU}\left(\boldsymbol{h}_t^L \boldsymbol{W}^P + \boldsymbol{b}^P\right)\boldsymbol{E}^T + \boldsymbol{b}^O\right) \tag{4.13}$$

gdje je $\boldsymbol{E} \in \mathbb{R}^{|\mathcal{V}| \times d}$ matrica ugradbenih vektora svih artikala.

4.2.2. Učenje modela

Za učenje modela primjenjena je tehnika modeliranja maskiranog jezika (engl. *masked-language modeling* – MLM) [3]. Ovom tehnikom ρ udio ulaznih artikala se maskira, tj. označava posebnim tokenom $[\text{mask}]$, a model onda uči tako da pogađa koji artikal je na mjestu maske uz korištenje poznatih artikala korisničke sjednice. Primjerice neka je ulaz lista $[v_1, v_2, v_3, v_4, v_5, v_6]$, a nakon primjene maske $[[\text{mask}]_1, v_2, v_3, [\text{mask}]_2, v_5, v_6]$ slijedi da su oznake (engl. *label*) $[\text{mask}]_1 = v_1$ i $[\text{mask}]_2 = v_4$. Funkcija gubitka (engl. *loss function*) je sljedeća:

$$\mathcal{L} = \frac{1}{|\mathcal{S}_u^m|} \sum_{v_m \in \mathcal{S}_u^m} -\log P\left(v_m = v_m^* \mid \mathcal{S}_u^{'}\right) \tag{4.14}$$

gdje je $\mathcal{S}_u^{'}$ maskirana verzija ulaznog niza artikala $\mathcal{S}_u$, $\mathcal{S}_u^m$ je skup maski, v_m^* predstavlja artikal koje se nalazi iza maske, a vjerojatnost P je definirana kao izraz (4.13).

Kako je zadatak predvidjeti sljedeći artikalu sjednici (izraz 2.1), testni skup podataka sadrži samo jedan $[\text{mask}]$ token na kraju sjednice.

Tehnikom MLM postiže se povećanje skupa podataka za učenje (engl. *data augmentation*) tako što je moguće izvesti $\binom{n}{k}$ primjeraka iz jednog primjerka za učenje,

gdje n predstavlja duljinu sjednice, a k je hiperparametar i predstavlja maksimalni broj $[\mathrm{mask}]$ tokena po primjerku. Proširivanjem podataka za učenje povećava se potrebno vrijeme za učenje, ali se postiže veći kapacitet modela.

5. Skupovi podataka

Postoji mali broj javnih skupova podataka s podacima o sjednicama korisnika najviše zbog čuvanja privatnosti korisnika. Dostupni podatkovni skupovi imaju jako malo informacija o artiklima i najčešće postoji samo podatak o identifikatoru artikla. Zbog navedenog ne može se empirijski ustanoviti koliko je sustav preporučivanja efikasan u praksi. Važno je naglasiti da nisu poznate okolnosti prikupljanja podataka te da postoji pristranost (engl. *bias*) zbog korištenja nekog sustava preporučivanja prilikom prikupljanja podataka. Pristranosti mogu dodatno pridonijeti promocije u e-trgovini što utječe na učenje pravilnosti. Slijedi da postoji mogućnost da su najbolji modeli oni koji najbolje rekonstruiraju model koji je bio u e-trgovini za vrijeme prikupljanja podataka [15].

U ovom radu korištena su dva skupa podataka: RSC15 i Diginetica[1] koje su autori koristili u [15]. Za provedbu eksperimenata u ovom radu uzet je jedan dio originalnih podataka. Iz tablice 5.1 je vidljivo da skup podataka RSC15 ima mnogo više korisničkih sjednica, ukupnih klikova uz podjednak vremenski raspon i broj artikala kao skup podataka Diginetica. Diginetica ima u prosjeku skoro jedan više klik po sjednici u odnosu na RSC15.

Tablica 5.1: Detalji o skupovima podataka uzetih za provedbu eksperimenata

Skupovi podataka	RSC15	Diginetica
Sjednice	1.6 milijuna	62 tisuće
Artikli	35.5 tisuće	34 tisuće
Klikovi	6.4 milijuna	301 tisuće
Vremenski raspon	31 dana	32 dana
Klikovi/sjednici	3.94	4.82

Da bi se eksperimenti mogli uspoređivati s prethodnim sličnim istraživanjima pro-

[1] `https://drive.google.com/drive/folders/1ritDnO_Zc6DFEU6UND9C8VCisT0ETVp5`

vedena je filtracija podataka kod oba skupa podataka. Uklonjene su sjednice s jednim artiklom te artikli koji se ne ponavljaju manje od 5 puta unutar skupa za učenje. Više uzastopnih klikova na iste artikle unutar jedne sjednice su zadržani. Iako na prvi mah ponovljena preporuka artikala nema smisla, takve preporuke mogu biti korisne iz korisničke perspektive, npr. kao podsjetnici [15].

5.1. Skup podataka RSC15

Podatkovni skup RSC15 sadrži korisničke klikove na artikle na nepoznatoj e-trgovini kroz period od 6 mjeseci. Bio je službeni skup podatka za natjecanje "RecSys Challenge 2015". Zadatak na natjecanju bio je predvidjeti hoće li korisnik (sjednica) nešto kupiti ili ne, te ako kupuje, koje artikle će kupiti. Podatkovni skup ima dva podskupa koji sadrže interakcije korisnika s artiklima e-trgovine. Svaki primjerak prvog podskupa sastoji se od sljedećih značajki: identifikator korisničke sjednice, vrijeme klika na artikal i kategoriju artikla. Drugi podskup sadrži događaje o kupnji, a sadrži sljedeće značajke: identifikator korisničke sjednice, vrijeme kada se kupnja dogodila, identifikator kupljenog artikla, te cijena i količina artikala.

Za provedbu eksperimenata uzet je samo prvi podskup podataka u kojem su ostavljeni samo identifikatori sjednice i artikala uz očuvanje redoslijeda artikala unutar sjednice i sjednica međusobno na osnovu vremenske značajke.

5.2. Skup podataka Diginetica

Skup podataka Diginetica korišten je za natjecanje "Personalized E-commerce Search Challenge" u sklopu konferencije CIKM iz 2016. godine [2]. Nastao je u kompaniji Diginetica, a uz podatke o korisničkim sjednicama i artiklima sadrži i podatke o upitima za pretraživanje. Skup podataka je anonimiziran tako da su *hashirani*: opisi i metapodaci proizvoda, upiti i pojmovi iz upita za pretraživanje. Na osnovu zadnjeg upita za pretraživanje u sjednici bilo je potrebno preporučiti listu artikala uz poznavanje svih artikala i upita u trenutnoj sjednici.

Za eksperimente u ovom radu i u radu [15] uklonjeni su podaci o upitima i korišteni su podaci koji su se sastojali o identifikatora sjednice i popisa artikala unutar sjednice. Sjednice i artikli su sortirani po vremenskoj značajci.

[2] https://competitions.codalab.org/competitions/11161#learn_the_details-overview

6. Evaluacija

Za provođenje eksperimenata korišten je kôd s GitHuba[1] za modele: SKNN, V-SKNN, STAN i GRU4Rec. Kôd modela BERT4Rec preuzet je GitHuba[2] te je dodatno prilagođen za SBRS jer je originalno napravljen za sustave preporučivanja *session-aware*.

Skupovi podataka sa svojstvima iz tablice 5.1 podjeljeni su na skupove za učenje i testiranje na način prikazan u tablici 6.1. Kod RSC15 skupa podataka skup za učenje sastojao se od sjednica kroz period od 30 dana i imao je približno 1.5 milijuna primjera, dok je skup za testiranje činio sjednice kroz jedan dan, a bilo ih je nešto manje od 88 tis. Kod skupa podataka Diginetica skup za učenje činile su sve korisničke sjednice u periodu 25 dana, a bilo ih je oko 48 tisuća, dok skup za testiranje imao je kroz period od 7 dana 14.5 tisuća sjednica.

Tablica 6.1: Detalji podjele na skup za učenje i testiranje

Skupovi podataka	RSC15	Diginetica
Skup za učenje		
Sjednice	1.5 milijuna	48 tisuće
Artikli	29.5 tisuće	34 tisuće
Klikovi	6 milijuna	233 tisuće
Vremenski raspon	30 dana	25 dana
Skup za testiranje		
Sjednice	88 tisuće	14.5 tisuće
Artikli	13.5 tisuće	18 tisuće
Klikovi	376 tisuće	68 tisuće
Vremenski raspon	1 dan	7 dana

[1] `https://github.com/rn5l/session-rec`
[2] `https://github.com/asash/BERT4rec_py3_tf2`

6.1. Evaluacijske metrike

Za evaluaciju rezultata korištene su najčešće metrike u sustavima preporučivanja čiji cilj je ocijeniti kvalitetu liste prporučenih artikala: HR@k, NDCG@k, POP@k i COV@k.

Metrika **HR@k** (engl. *hit ratio*) predstavlja udio pogodaka, tj. udio predikcija algoritma u kojima je ciljni artikal bio preporučen unutar liste od k artikala. Ova metrika ne uzima u obzir rang ciljnog artikla unutar liste preporučenih artikala.

Normalizirani diskontirani kumulativni dobitak (engl. *Normalized Discounted Cumulative Gain – NDCG*) je metrika koja uzima u obzir rang ciljnog artikla unutar liste preporučenih artikala. Iznos ove metrike je veći što algoritam daje niži rang artiklima koji su relevantni za korisnika.

Za definiciju metrike potrebno je prvo definirati što je kumulativni dobitak i diskontirani kumulativni dobitak. Kumulativni dobitak (engl. *Cumulative Gain– CG*) se računa kao:

$$CG = \sum_{i=1}^{k} rel_i \tag{6.1}$$

gdje je rel_i relevantnost artikla na poziciji i u preporučenoj listi duljine k. Diskontirani kumulativni dobitak (engl. *Discounted Cumulative Gain – DCG*) uzima u obzir redoslijed za razliku od CG, a dobiva se kao:

$$DCG = \sum_{i=1}^{k} \frac{rel_i}{\log_2 (i+1)} \tag{6.2}$$

Normalizirani diskontirani kumulativni dobitak je:

$$NDCG = \frac{DCG}{iDCG}, \tag{6.3}$$

gdje je $iDCCG$ je diskontirani kumulativni dobitak idealnog poretka artikala [1]. Kako je zadatak predvidjeti sljedeći artikal (formula 2.1), relevantnost artikla na poziciji i je 1 ako je taj artikal sljedeći artikal, a za sve ostale pozicije vrijednost je 0. Slijedi da je $iDCCG = 1$ koji se dobiva kad je ciljni artikl prvi u preporučenoj listi artikala.

Pokrivenost (engl. *coverage*) može se definirati na dva načina:

1. postotak artikala koje je sustav barem jednom preporučio,
2. postotak artikala koje sustav može preporučiti [6].

U ovom radu korišten je prva definicija pokrivenosti koja se još zove pokrivenost kataloga (engl. *catalog coverage*). Neka je I_L^j skup artikala koji se nalazi u listi L duljine k

koja je preporučena za j-tu predikciju modela. N neka označava ukupan broj predikcija modela, a I skup svih artikala iz skupa za učenje. Formula pokrivenosti kataloga **COV@k** je [5]:

$$COV@k = \frac{|\cup_{j=1...N} I_L^j|}{|I|}. \tag{6.4}$$

Poželjnije je da model ima veću pokrivenost, ali nikad da veća pokrivenost bude na štetu preciznosti algoritma. Primjerice ako postoje neki artikli koji nisu zanimljivi nijednom korisniku onda dobar model nikad neće preporučivati takve artikle što vodi do manje pokrivenosti ali veće preciznosti [6]. Manje vrijednosti pokrivenosti kataloga pokazuju da model ima tendenciju preporučivati isti skup artikala što može ukazivati da model ne prilagođava preporuke ovisno o kontekstu (korisničkoj sjednici).

Razinu pristranosti modela preporučivanju popularnih artikala mjeri popularnost (engl. *popularity*). Za metriku **POP@k** prvo se izračuna skaliranje min-maks broja ponavljanja artikala u skupu za učenje. Skaliranje min-maks za artikl se računa tako da se broj ponavljanja artikla oduzme od najmanjeg broja ponavljanja nekog artikla, a zatim podijeli sa razlikom maksimalnog i minimalnog broja ponavljanja. Skalirana vrijednost ponavljanja artikla u skupu za učenje je koeficijent popularnosti tog artikla. Transformirane vrijednosti su u rasponu od $[0, 1]$, a pridružuju se svakom artiklu u skupu za učenje. Tako, najpopularniji artikal ima vrijednost 1, a najmanje popularan artikal ima vrijednost 0. Za svaku preporučenu listu duljine k modela računa se prosjek koeficijenata popularnosti artikala. POP@k predstavlja prosječni koeficijent popularnosti na svim predikcijama modela [15].

Prosječni recipročni rang (engl. *mean reciprocal rank – MRR*) se računa kao:

$$MRR@k = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rang_i} \times 1_k(t) \tag{6.5}$$

gdje je $|Q|$ broj predikcija modela, $rang_i$ pozicija ciljnog artikla unutar liste preporučenih artikala, a funkcija $1_k(t)$ je 1 ako se ciljni artikal t nalazi u k prvih pozicija unutar preporučene liste, a 0 inače [17].

6.2. Implementacijski detalji

Za izvršavanje modela SKNN, V-SKNN, STAN i GRU4Rec korišteni su hiperparametri dobiveni u [15].

6.2.1. Model SKNN

Za model **SKNN** *sample_size* je podskup svih sjednica za učenje iz kojih se odabire *number_of_neighbors* najsličnijih sjednica iz kojih se preporučuju sljedeći artikli. Za skup podataka RSC15 koristi se Jaccardov koeficijent sličnosti (formula 3.2), a za skup podataka Diginetica sličnost kosinusa (formula 3.3).

Tablica 6.2: Hiperparametri modela SKNN

Hiperparametri	Skup podataka	
	RSC15	Diginetica
number_of_neighbors	500	100
sample_size	10000	500
similarity	Jaccard	cosine

6.2.2. Model V-SKNN

Model V-SKNN u odnosu na SKNN uvodi nove parametre: *weighting*, *weighting_score* i *IDF_weighting*. Parametar *weighting* koristi se za ponderiranje artikala unutar trenutne sjednice pri izračunu sličnosti sa susjednom sjednicom, dok parametar *weighting_score* je funkcija $w(v, s)$ iz formule 3.4 i koristi se za ponderiranja skora artikala iz susjedne sjednice. Parametar *IDF_weighting* koristi se da bi se pojačao (smanjio) skor onih artikala iz susjedne sjednice koji se rjeđe (češće) pojavljuju u skupu podataka. Svi hiperparametri se nalaze u tablici 6.3. Funkcija div iz tablice 6.3 se računa

Tablica 6.3: Hiperparametri modela V-SKNN

Hiperparametri	Skup podataka	
	RSC15	Diginetica
number_of_neighbors	100	500
sample_size	1000	5000
weighting	quadratic	
weighting_score	quadratic	div
IDF_weighting	False	5

kao:

$$w(v, s) = div(v_i, s) = \frac{i}{l(s)}, \tag{6.6}$$

gdje je i pozicija artikla v unutar sjednice, a $l(s)$ duljina trenutne sjednice.

6.2.3. Model STAN

Model STAN uz standardne parametre koje koriste modeli temeljeni na k najbližih sjednica (*number_of_neighbors* i *sample_size*) dodaje njemu specifične parametre (tablica 6.4): *lambda_spw*, *lambda_snh* i *lambda_inh*. Parametar *lambda_spw* je λ_1 parametar iz formule 3.6, parametar *lambda_snh* je λ_2 parametar iz formule 3.8, a parametar *lambda_inh* je λ_3 parametar iz formule 3.10.

Tablica 6.4: Hiperparametri modela STAN

Hiperparametri	Skup podataka	
	RSC15	Diginetica
number_of_neighbors	1000	500
sample_size	10000	
lambda_spw	0.00001	1.225
lambda_snh	10	20
lambda_inh	2	4.9

6.2.4. Model GRU4Rec

Model GRU4Rec na oba skupa podataka daje najbolje rezultate s TOP1-MAX funkcijom gubitka. Također, najbolje rezultati dobivanju se korištenjem *constrained_embedding* što znači da se matrica težina izlaznog sloja koristi kao matrica ugradbenih vektora za artikle. Takva arhitektura je najoptimalnija što se tiče korištenja memorije.

Tablica 6.5: Hiperparametri modela GRU4Rec

Hiperparametri	Skup podataka	
	RSC15	Diginetica
loss	TOP1-MAX	
final_act	linear	
dropout_p_hidden	0.3	0.4
learning_rate	0.04	0.05
constrained_embedding	True	

6.2.5. Model BERT4Rec

Za model BERT4Rec za optimizaciju hiperparametara na oba podatkovna skupa podataka odabrana je 5-struka unakrsna validacija (engl. *k-fold cross validation*). Provedena je optimizacija 2 hiperparametra: maksimalna duljina sekvence i veličina mini grupe. Maksimalna duljina sekvence t (*max_seq_length*) određuje maksimalni broj ulaza u model (slika 4.3). Tako, model uzima $t - 1$ prethodnih artikala kako bih preporučio listu sljedećih artikala. Vrijednosti koje su isprobane za maksimalnu duljinu sekvence su: 10, 30 i 50. Maksimalna duljina sekvence je znatno manja nego duljina sekvenci u području NLP-a jer su duljine sjednica manje (tablica 5.1) od broja riječi koje BERT prima na ulazu u području NLP-a. Za veličine mini grupe (*batch_size*) isprobane su vrijednosti: 16 i 32. Veličina mini grupe određuje broj primjeraka nakon kojih će se provesti ažuriranja težina modela. Model s najboljim hiperparametrima odabran je na temelju najbolje prosječne vrijednosti NDCG@20 na 5 preklopa.

Ostali hiperparametri preuzeti su iz rada [18] i nad njima nije provedena iscrpna pretraga zbog velikog broja potencijalnih modela, a ograničenih resursa. Hiperparametri s kojima su dobiveni najbolji rezultati prikazani su u tablici 6.6. Od ostalih hiper-

Tablica 6.6: Hiperparametri s kojima su dobiveni najbolji rezultati modela BERT4Rec

Hiperparametri	Skup podataka	
	RSC15	Diginetica
Model		
batch_size	16	32
max_seq_length	10	
dim	100	
num_train_steps	400 000	
learning_rate	$1e-4$	
AdamW		
β_1	0.9	
β_2	0.999	
ϵ	$1e-6$	
Podaci		
dupe_factor	10	
mask_prob	1.0	
prop_sliding_window	0.5	
max_pred_seq	10	

parametara specifičnih za model *dim* predstavlja parametar d modela, *num_train_steps* označava ukupan broj ažuriranja težina, a *learning_rate* je stopa učenja. Vrijednost parametra *dim* postavljena je na 100 kako bi se moglo uspoređivati s modelom GRU4Rec s istom veličinom skrivenog sloja.

Za metodu optimizacije parametara mreže korišten je AdamW.

Za pripremu podataka korišteni su sljedeći parametri: *dupe_factor*, *mask_prob*, *prop_sliding_window* i *max_pred_seq*. *dupe_factor* označava koliko puta će se provoditi tehnika MLM nad svakim od primjeraka skupa za učenje. Parametar *mask_prob* predstavlja vjerojatnost da će biti provedena teknika MLM nad primjerkom iz skupa za učenje. *prop_sliding_window* je parametar koji se koristi kad je neka sjednica dulja od *max_seq_length*, pa se koristi pomak prozora koji je jednak *prop_sliding_window* $\times$ *max_seq_length*. Parametar *max_pred_seq* određuje maksimalni broj $[mask]$ tokena unutar sjednice.

Broj $[mask]$ tokena unutar sjednice (isječak kôda 6.1, preuzeto sa [3]) je uvijek između 1 (jer je *mask_prob* $= 1$) i *max_pred_seq*.

Isječak kôda 6.1: Prikaz izračuna broja *mask* tokena

```
num_to_predict = min(max_predictions_per_seq,
                     max(1, int(round(len(tokens) * masked_lm_prob))))
```

6.3. Rezultati

Rezultati po podatkovnim skupovima su prikazani u tablici 6.7. Modeli po podatkovnim skupovima su poredani po MRR@20 vrijednosti od boljih k lošijima. Najbolji rezultat po metrici je podebljan.

Zanimljivo je da su osnovni modeli kompetitivni ili bolji od neuronskih modela, iako su modeli manjeg kapaciteta. Uz to, osnovni modeli nisu modeli crne kutije (engl. *black box*) što znači da se njihove preporuke mogu objasniti za razliku od neuronskih modela. Iz rezultata je vidljivo da model GRU4Rec dominira u metrikama COV@20 i POP@20 što znači da model uspijeva nalaziti nepopularne artikle koji su relevantni korisnicima, a da pritom ne gubi na rangiranju ciljnog artikla. Model BERT4Rec je podjednak s ostalim modelima na skupu podataka RSC15, osim u metrikama: COV@20 i POP@20, a lošiji je u odnosu na ostale modele na skupu podataka Diginetica. Iako model s najvećim kapacitetom i dvosmjernim učenjem ne uspijeva biti bolji od puno

[3] https://github.com/asash/BERT4rec_py3_tf2

Tablica 6.7: Rezultati svih modela po podatkovnim skupovima

Ev. metrike	MRR@20	NDCG@20	HR@20	COV@20	POP@20
RSC15					
STAN	**0.3403**	**0.4247**	**0.7062**	0.6479	0.0690
BERT4Rec	0.3370	0.4012	0.6135	0.0332	0.0748
V-SKNN	0.3310	0.4138	0.6908	0.6132	0.0639
SKNN	0.3301	0.4109	0.6823	0.6478	0.0709
GRU4Rec	0.3190	0.4040	0.6896	**0.7237**	**0.0308**
Diginetica					
STAN	**0.1941**	0.2493	0.4397	0.8276	0.0944
V-SKNN	0.1921	0.2468	0.4354	0.8592	0.0852
SKNN	0.1898	0.2467	0.4423	0.8280	0.0978
GRU4Rec	0.1865	**0.2510**	**0.4793**	**0.8607**	**0.0639**
BERT4Rec	0.1643	0.1942	0.2981	0.0328	0.4425

jednostavnijih modela. Možda model BERT4Rec ne može iskoristiti puni potencijal samopozornosti zbog kratkih sjednica u SBRS problemu (RSC15 i Diginetica < 5 artikala u prosjeku po sjednici) jer osnovni model ima puno veći broj ulaznih riječi (maksimum 512). Također potencijalno modelu BERT4Rec je potrebno više podataka jer je osnovni model BERT prednaučen na velikom skupu podataka.

Iz tablice 6.7 je vidljivo da model V-SKNN ima manju vrijednost metrike POP@20 od modela SKNN na oba skupa podataka, a izraženije na skupu podataka Diginetica gdje je model V-SKNN koristio ponderiranje skora na temelju broja pojavljanja artikla u sjednicama (vidi parametar *IDF_weighting* iz tablice 6.3). Razlog ovakvih rezultata može biti da ponderiranje skora artikala smanjuje tendenciju modela prema popularnim artiklima i povećava pokrivenost kataloga.

Na temelju rezultata metrike MRR@20 na skupu podataka RSC15 može se zaključiti da modeli prosječno preporučuju artikle oko trećeg mjesta liste za preporučivanje, a kod skupa podataka Diginetica na diobi petog i šestog mjesta. Uz navedeno zapažanje treba uzeti u obzir da modeli na skupu podataka RSC15 u 30 % slučajeva ne preporučuju artikal unutar liste od 20 artikala (HR@20 ≈ 0.7), a kod skupa podataka Diginetica oko 55 % slučajeva.

6.4. Usporedba rezultata s drugim radovima

Rezultati dobiveni u radu [15] prikazani su u tablici 6.8. Ovi rezultati su dobiveni uprosječavanjem na 5 dijelova svakog od skupova podataka. Također, rezultati su dobiveni drugačijim modeliranjem učenja i testiranja. Sjednica duljine t podjeljena je na $t-1$ primjerak gdje su primjerci dobiveni na sljedeći način: $[v_1] \rightarrow v_2$, $[v_1, v_2] \rightarrow v_3$, itd. Dakle, kontekst trenutne sjednice u početku čini jedan artikal na osnovu kojeg se predviđa sljedeći za koji se računa gubitak. Nakon predviđanja, sljedeći artikal postaje dio konteksta trenutne sjednice na osnovu kojeg se dalje predviđa sljedeći artikal i tako dok se ne dođe do kraja sjednice.

Tablica 6.8: Rezultati svih modela po podatkovnim skupovima u radu [15]

Ev. metrike	MRR@20	HR@20	COV@20	POP@20
RSC15				
STAN	**0.2933**	**0.6656**	0.6828	0.0773
V-SKNN	0.2872	0.6512	0.6333	0.0777
GRU4Rec	0.2826	0.6480	**0.7482**	**0.0294**
SKNN	0.2620	0.5996	0.6099	0.0796
Diginetica				
STAN	**0.1828**	**0.4800**	0.9161	0.0964
V-SKNN	0.1784	0.4729	0.9419	0.0840
SKNN	0.1714	0.4748	0.8701	0.1026
GRU4Rec	0.1644	0.4639	**0.9498**	**0.0567**

U tablici 6.8 nema rezultata modela BERT4Rec jer on do sada nije primjenjen u sustavu preporučivanja SBRS. U rezultatima nema rezultata mjere NDCG@20 koja nije korištena u radu [15]. Dobiveni rezultati za metrike rangiranja u radu [15] su lošiji od dobivenih rezultata u ovom radu, a to vjerojatno zbog drugačijeg definiranog zadatka zbog kojeg sjednice u ovom radu su imale duži kontekst u odnosu na sjednice iz rada [15]. Metrike pokrivenosti su bolje u radu [15] iz razloga što je provedeno više predikcija i samim time preporučeno više različitih artikala. U oba rada najbolji rezultati za metriku MRR@20 postižu se osnovnim modelom STAN, dok najbolji rezultati za metrike COV@20 i POP@20 postiže model GRU4Rec na oba skupa podataka u oba rada.

6.5. Pokušaji poboljšanja modela BERT4Rec

Nakon dobivenih slabijih rezultata s modelom BERT4Rec, osobito na podatkovnom skupu Diginetica, pokušalo se provesti dodatne eksperimente kako bi se unaprijedili dobiveni rezultati. Autori u radu [9] dobili najbolje rezultate sa samo jednim slojem GRU unutar modela GRU4Rec (slika 4.1) i bez dodatnih potpuno povezanih unaprijednih slojeva nakon GRU sloja. Pretpostavili su da je razlog što najbolje rezultate dobijaju sa najjednostavnijom arhitekturom taj da su sjednice prekratke i da je jedan GRU sloj dovoljan da se dobije dobra reprezentacija trenutne sjednice na osnovu koje se preporučuje.

Pokušaji poboljšanja su provedeni na skupu podataka Diginetica. Zbog prije navedenog, poboljšanja su jednim djelom išli u smjeru pojednostavljivanja modela BERT4Rec. Jedan pojednostavljeni model koristio je 1 glavu samopozornosti i 1 transformerski sloj, a drugi je koristio 2 glave samopozornosti i 1 transformerski sloj. Kod oba modela ostali parametri bili su jednaki onima iz tablice 6.6.
Drugi dio poboljšanja odnosio se na fino podešavanje. Kod finog podešavanja modela sve sjednice iz skupa za učenje imale su masku na zadnjem mjestu u sjednici kako bi se model što bolje prilagodio zadatku na testnom skupu podataka. Korišten je model s hiperparametrima iz tablice 6.6 s već naučenim težinama.

Tablica 6.9: Rezultati pokušaja poboljšanja modela BERT4Rec na skupu podataka Diginetica

Ev. metrike	NDCG@20	HR@20	COV@20	POP@20
Diginetica				
1 glava, L=1	0.1755	0.2774	0.0236	0.4447
2 glave, $L = 1$	0.1603	0.2652	0.0217	0.3568
fino podešavanje	0.1766	0.2846	0.0255	0.4911

Kako je vidljivo iz tablice 6.9 pokušaji poboljšanja nisu doveli do boljih rezultata za model BERT4Rec u odnosu na dobivene rezultate ovog modela na skupu podataka Diginetica iz tablice 6.7.

7. Poboljšanja i budući smjerovi razvoja

Razmatranje više svojstava o proizvodima kao što su metapodaci (cijena, opis proizvoda, recenzija od usera, ocjene), slika, itd. je posebno važno kod problema hladnog starta (engl. *cold start problem*) kad ne postoje interakcije s tim proizvodom, pa metapodaci se mogu iskoristiti da se nađu slični proizvode. Navedeno je implementirano u [8] i postigli su se poboljšani rezultati.

U budućnosti može se napraviti iskorak korištenjem podataka o kontekstu korisnika u kojem se nalazi kao što su vrijeme, godišnje doba, lokacija, vrijeme i nedavni trend popularnosti [10].

Korištenje više vrsta korisničkih interakcija koje nisu samo klikovi: dodavanje u košaricu, kupnja artikla, vrijeme zadržavanja na proizvodu, i dr. Uključivanjem novih interakcija dolazi do izazova ispravnog omjera vrednovanja interakcija, npr. koliko je kupnja artikla jači signal od samo gledanja artikla. Također, vrednovanje interakcija ne može biti jednako za sve korisnike jer primjerice neki korisnici mogu biti neodlučniji pa češće dodavati artikal u košaricu bez da kupe artikal.

Jedan od smjerova razvoja je uključivanje implicitne negativne povratne informacije (engl. *feedback*), a ne samo korištenje implicitne pozitivne povratne informacije (npr. klikovi). Primjer implicitne negativne informacije može biti ne klikanje na artikal koji je preporučen visoko na listi preporuka. Ovdje je također izazov modeliranja implicitne negativne informacije.

Postoji i mogućnost razvijanja novih funkcija gubitka koje su prilagođene problemu rangiranja.

8. Zaključak

Sustavi preporučivanja temeljeni na korisničkim sjednicama (SBRS) preporučuju sljedeće artikle samo na temelju informacija iz trenutne sjednice te podataka o korisničkim sjednicama drugih korisnika. Cilj ovakvog sustava preporučivanja je da proba primjenom naučenih obrazaca ponašanja korisnika i na temelju trenutne sjednice predvidjeti sljedeći artikl na koji će kliknuti anonimni korisnik. Anonimnom korisniku se preporučuje lista artikala koji su najizgledniji da će ih korisnik kliknuti u sljedećem koraku.

U ovom radu detaljno su objašnjeni modeli koji daju najbolje rezultate na ovom problemu, a zatim su provedeni eksperimenti na skupovima podataka RSC15 i Diginetica. Ovi podatkovni skupovi sadrže jedino informaciju o kliknutim artiklima unutar korisničkih sjednica. U ovom radu korišteni su osnovni modeli: SKNN, V-SKNN i STAN te neuronski modeli: GRU4Rec i BERT4Rec.

Model BERT4Rec korišten je u sustavima preporučivanja (engl. *session-aware*) i prvi puta u ovom radu je prilagođen i primjenjen na SBRS problemu. Model BERT4Rec daje podjednake rezultate s ostalim modelima na skupu podataka RSC15, dok je najlošiji na skupu podataka Diginetica. Također, daje najlošije rezultate za metrike: COV@20 i POP@20 što je potrebno dodatno istražiti.

Model GRU4Rec daje najbolje rezultate za metrike COV@20 i POP@20 na oba skupa podataka. Provedenim eksperimentima zaključeno je da osnovni modeli iako jednostavni i dalje daju podjednake ako ne i bolje rezultate od složenijih neuronskih modela.

Budućnost sustava SBRS svakako je u uključivanju dodatnih podataka o artiklima kao i razvijanje modeliranja dodatnih korisničkih interakcija sa sustavom sve u cilju kako bi se povećala uspješnost navedenih modela.

LITERATURA

[1] Pranay Chandekar. Evaluate your recommendation engine using ndcg. https://towardsdatascience.com/evaluate-your-recommendation-engine-using-ndcg-759a851452d1, 13. 1. 2020. Pristupljeno: lipanj 2022.

[2] KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, i Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014. URL http://arxiv.org/abs/1409.1259.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, i Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL https://arxiv.org/abs/1810.04805.

[4] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, i Gautam Shroff. Sequence and time aware neighborhood for session-based recommendations: Stan. U *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, stranica 1069–1072, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361729. doi: 10.1145/3331184.3331322. URL https://doi.org/10.1145/3331184.3331322.

[5] Mouzhi Ge, Carla Delgado-Battenfeld, i Dietmar Jannach. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. U *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, stranica 257–260, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589060. doi: 10.1145/1864708.1864761. URL https://doi.org/10.1145/1864708.1864761.

[6] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, i John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*,

22(1):5–53, jan 2004. ISSN 1046-8188. doi: 10.1145/963770.963772. URL https://doi.org/10.1145/963770.963772.

[7] Balázs Hidasi i Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. *CoRR*, abs/1706.03847, 2017. URL http://arxiv.org/abs/1706.03847.

[8] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, i Domonkos Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. U *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, stranica 241–248, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340359. doi: 10.1145/2959100.2959167. URL https://doi.org/10.1145/2959100.2959167.

[9] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, i Domonkos Tikk. Session-based recommendations with recurrent neural networks, 2015. URL https://arxiv.org/abs/1511.06939.

[10] D. Jannach, Bamshad Mobasher, i Shlomo Berkovsky. Research directions in session-based and sequential recommendation. *User Modeling and User-Adapted Interaction*, 30:609 – 616, 2020.

[11] Dietmar Jannach i Malte Ludewig. When recurrent neural networks meet the neighborhood for session-based recommendation. U *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, stranica 306–310, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346528. doi: 10.1145/3109859.3109872. URL https://doi.org/10.1145/3109859.3109872.

[12] Dietmar Jannach i Markus Zanker. *Collaborative Filtering: Matrix Completion and Session-Based Recommendation Tasks: Algorithms, Practical Challenges and Applications*, stranice 1–34. 11 2018. ISBN 978-981-327-534-8. doi: 10.1142/9789813275355_0001.

[13] Dietmar Jannach, Lukas Lerche, i Michael Jugovac. Adaptation and evaluation of recommendations for short-term shopping goals. U *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, stranica 211–218, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336925. doi: 10.1145/2792838.2800176. URL https://doi.org/10.1145/2792838.2800176.

[14] Malte Ludewig i Dietmar Jannach. Evaluation of session-based recommendation algorithms. *CoRR*, abs/1803.09587, 2018. URL http://arxiv.org/abs/1803.09587.

[15] Malte Ludewig, Noemi Mauro, Sara Latifi, i Dietmar Jannach. Empirical analysis of session-based recommendation algorithms. *CoRR*, abs/1910.12781, 2019. URL http://arxiv.org/abs/1910.12781.

[16] Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, i Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013. URL http://arxiv.org/abs/1310.4546.

[17] János Moldvay. Evaluating recommender systems. https://www.blabladata.com/2014/10/26/evaluating-recommender-systems/, 26. 10. 2014. Pristupljeno: lipanj 2022.

[18] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, i Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. *CoRR*, abs/1904.06690, 2019. URL http://arxiv.org/abs/1904.06690.

Sustavi preporučivanja temeljeni na korisničkim sjednicama

Sažetak

Sustavi preporučivanja temeljeni na korisničkim sjednicama (engl. *Session-based recommender systems–SBRS*) pružaju preporuke isključivo na temelju interakcija korisnika u trenutnoj sjednici i ne zahtijevaju postojanje korisničkih profila ili njihovih cjelokupnih povijesnih aktivnosti. Cilj takvih sustava je da na temelju niza kliknutih artikala u sjednici preporučuje artikle koji su najizgledniji da će na jednog od njih kliknuti korisnik u sljedećem koraku. U ovom diplomskom radu opisani su i primjenjeni modeli koji daju najbolje rezultate na navedenom problemu u području e-trgovine. Model BERT4Rec, originalno objavljen za sustave preporučivanja koji koriste sve prošle sjednice istog korisnika (engl. *session-aware recommendation systems*), detaljno je objašnjen i prilagođen navedenom problemu. Svi modeli evaluirani su uobičajenim tehnikama za sustave preporučivanja i međusobno uspoređeni. Na kraju rada navedeni su potencijalni smjerovima razvoja ovog područja.

Ključne riječi: sustavi preporučivanja, sjednica, klikovi, e-trgovina, GRU4Rec, KNN, BERT4Rec

Session-based recommender systems

Abstract

Session-based recommender systems (SBRS) provide recommendations based solely on user interactions in current session and do not require the existence of user profiles or their entire historical activities. Based on a series of clicked items in a session, the goal of such systems is to recommend items that are most likely to be clicked on by the user in the next step. In this master thesis, the models that give the best results for the mentioned problem in the field of e-commerce are described and applied. The BERT4Rec model, originally published for session-aware recommender systems, is explained in detail and adapted to the given problem. All models were evaluated using common techniques for evaluating recommender systems and compared with each other. The potential directions of the development of this area are presented at the end of the thesis.

Keywords: recommender systems, session, click, e-commerce, GRU4Rec, KNN, BERT4Rec