Music Emotion Classification Using Lyrics

Projekat u okviru kursa Računarska inteligencija

Julijana Jevtić 25/2020
Jelena Milošević 69/2020

Septembar, 2025

Univerzitet u Beogradu, Matematički fakultet

Sadržaj

1 Uvod

Klasifikacija emocija iz muzičkih tekstova predstavlja zadatak u oblasti *obrade prirodnog jezika (NLP)* i *mašinskog učenja*. Cilj je analizirati lirski sadržaj tekstova pesama i odrediti kojoj emociji pripada, na primer: sreća, tuga, ljutnja ili ljubav. Emocije u muzici imaju veliki uticaj na slušaoce, a njihovo automatsko prepoznavanje može se primeniti u preporučivačkim sistemima, analizi muzičkih trendova i personalizaciji muzičkih servisa.

Ovaj zadatak kombinuje:

- **Analizu teksta** – razumevanje značenja reči, izraza i konteksta u lirici pesme.
- **Modelovanje sekvenci** – hvatanje strukture i emotivnog toka koji se razvija kroz stihove pesme.

Automatska klasifikacija emocija u muzici postaje sve važnija jer omogućava efikasnije pretraživanje muzičkih baza, kreiranje plejlista zasnovanih na raspoloženju, kao i dublje razumevanje povezanosti između teksta i ljudskih emocija.

2 Priprema podataka

Prikupljanje podataka:

Iz obimnog skupa podataka sa preko 500,000 primeraka koristile smo čitave tekstove pesama uz procenjene emocije radi treniranja naših modela za klasifikaciju. [1]

Preprocesiranje podataka:

Preprocesiranje tekstualnih podataka sprovedeno je kako bi se obezbedila konzistentnost i smanjio šum u korpusu pesama. Glavni koraci uključuju:

- **normalizaciju teksta -** tokeni su konvertovani u mala slova, uklonjeni su linkovi, anotacije u uglastim zagradama, kao i specijalni karakteri koji nisu deo engleskog alfabeta),
- **zamenu slenga i kontrakcija -** korišćen je rečnik čestih internet izraza i muzičkog slenga, kao i kontrakcija za semantički bogatije i formalizovanije predstavljanje teksta,
- **korišćenje regex pravila -** regularni izrazi korišćeni su za dodatnu ekspanziju skraćenica,
- **čišćenje i tokenizaciju -** višestruki razmaci zamenjeni su jednim, a nepotrebni znakovi uklonjeni.

Ovakav proces omogućava bolje generalizovanje modela, smanjenje broja različitih varijanti iste reči i lakše prepoznavanje emocionalnih obrazaca u tekstu pesama.

Priprema i podela podataka:

Za treniranje modela mašinskog učenja, neophodno je da se početni skup podataka podeli na disjunktne podskupove. U našem radu koristimo sledeću podelu: **trening skup** (80% podataka) - za učenje parametara modela; **validacioni skup** (10% podataka) -hiperparametara i prevenciju overfitting-a; **test skup** (10% podataka) - za konačnu evaluaciju performansi modela.

Podela je izvedena uz pomoć metode *stratifikovane nasumične deobe*, kako bi se očuvala raspodela klasa u svakom podskupu i time izbegla pristrasnost prema dominantnim klasama.

Čuvanje podeljenih podataka:

S obzirom na to da upoređujemo performanse različitih modela (*LSTM* i *Transformer*), neophodno je da treniranje i evaluacija svih arhitektura budu izvršene nad identičnim podacima. Zbog toga se jednom generisani splitovi eksplicitno čuvaju u odvojenim CSV fajlovima zajedno sa enkoderom labela. Ovim postupkom se eliminiše varijabilnost koja bi mogla nastati ponovnim nasumičnim deljenjem podataka.

Hiperparametri specifični za sekvencijalne modele:

- maksimalnu dužinu sekvence koja predstavlja gornju granicu broja tokena u jednoj instanci (obezbeđuje da svi ulazi imaju istu dimenziju, što je neophodan uslov za mini-batch treniranje)

- veličinu rečnika koji se konstruiše na osnovu učestalosti tokena u trening skupu, a reči koje se ne nalaze u prvih V najfrekventnijih mapiraju se u poseban `UNK` token. (postiže kompromis između izražajne moći modela i njegove računarske efikasnosti)

3 Opis rešenja

Za rešavanje ovog problema ispitani su modeli zasnovani na sekvencijalnoj obradi podataka: **Long Short-Term Memory (LSTM)** i **Transformer arhitektura**.

3.1 Long Short-Term Memory (LSTM) model

LSTM je posebna vrsta rekurentne neuronske mreže (RNN) koja je dizajnirana za obradu sekvenci i rešavanje problema dugoročnih zavisnosti. Tradicionalne RNN mreže često imaju poteškoće sa učenjem kada su zavisnosti u tekstu udaljene, jer gradijenti vremenom nestaju. LSTM uvodi *gate* mehanizme koji omogućavaju selektivno čuvanje i zaboravljanje informacija.

3.1.1 Kako LSTM obrađuje tekst

1. Zaboravlja nevažne informacije iz prethodnih stihova.
2. Dodaje nove relevantne informacije koje opisuju emociju u trenutnom kontekstu.
3. Ažurira izlaz, šaljući dalje samo značajne podatke za klasifikaciju emocije.

3.1.2 Varijante i unapređenja LSTM arhitekture

Postoje različite varijante LSTM modela koje omogućavaju bolje učenje dugoročnih zavisnosti i poboljšanu obradu sekvenci [3]:

- **Standard (Vanilla) LSTM:** osnovni LSTM model sa **jednim slojem** koji procesuira sekvencu korak po korak. Tipične primene uključuju predikciju vremenskih serija i jednostavnu klasifikaciju sekvenci.
- **Stacked / Deep LSTM:** LSTM arhitektura sa **više slojeva**. Povećava kapacitet učenja i omogućava modelu da uči složenije obrasce. Primene: prepoznavanje govora, analiza video sadržaja.
- **Bidirectional LSTM (BiLSTM / BLSTM):** procesira sekvencu **u oba smera**, napred i nazad, što omogućava korišćenje konteksta iz prošlosti i budućnosti. Prednosti: preciznije predikcije i bolje razumevanje konteksta. Primene: prepoznavanje entiteta u tekstu (NER), mašinsko prevođenje.
- **LSTM sa Attention mehanizmom:** kombinuje LSTM sa mehanizmom pažnje koji fokusira model na relevantne delove ulaza prilikom predikcije. Prednosti: bolja obrada dugih sekvenci i dinamički fokus na kontekst. Primene: mašinsko prevođenje, sažimanje teksta.
- **Encoder-Decoder LSTM:** arhitektura sa dva LSTM modela: enkoder sažima ulaznu sekvencu, a dekoder generiše izlaznu sekvencu. Prednosti: pogodno za zadatke sa promenljivom dužinom izlaza. Primene: prevođenje, generisanje sekvenci, sažimanje teksta.

3.1.3 Primena LSTM modela

LSTM modeli se široko primenjuju u različitim oblastima zbog svoje sposobnosti da uče dugoročne zavisnosti u sekvencijalnim podacima [3]:

- **Obrada prirodnog jezika (NLP):** analiza sentimenta, klasifikacija teksta, prepoznavanje entiteta, mašinsko prevođenje.
- **Generisanje teksta i jezika:** generisanje muzičkih tekstova, dijaloga i automatsko pisanje.
- **Prepoznavanje govora:** pretvaranje govora u tekst, razumevanje konteksta u audio sekvencama.
- **Obrada vremenskih serija:** predviđanje cena akcija, ekonomskih indikatora, vremenskih uslova.
- **Biomedicina i biosignali:** analiza EKG signala, EEG podataka, detekcija bolesti iz sekvencijalnih podataka.
- **Računarska vizija:** opisivanje slika (image captioning), analiza video snimaka, praćenje objekata.

- **Kontrola i robotika:** predviđanje akcija, modelovanje kretanja robota ili vozila.
- **Cloud computing i resursna optimizacija:** predviđanje opterećenja serverskih centara radi efikasnijeg skaliranja resursa i smanjenja potrošnje energije [4].
- **Primena u akustičkom modelovanju i prepoznavanju govora:** Deep LSTM i LSTMP (LSTM sa recurrent projection layer) arhitekture omogućavaju postizanje state-of-the-art performansi u velikim vokabularima, nadmašujući DNN i standardne RNN modele [5].

3.1.4 Primena u klasifikaciji emocija

U ovom projektu, LSTM se koristi za analizu sekvenci reči u lirici. Model uspeva da uhvati emotivni kontekst pesme – na primer, uočava da ponavljanje reči *"goodbye"* i *"cry"* upućuje na tugu, dok kombinacija izraza *"love"*, *"forever"* i *"heart"* nagoveštava emociju ljubavi.

3.1.5 Teorijska osnova i istorijat LSTM-a

Problem tradicionalnih rekurentnih neuronskih mreža (RNN) je *nestajanje* ili *eksplozija gradijenata* tokom učenja sekvenci korišćenjem metoda poput *back-propagation through time* (BPTT) ili *real-time recurrent learning* (RTRL). Kada se greška propagira unazad kroz mnogo vremenskih koraka, gradijenti se mogu eksponencijalno smanjivati ili povećavati u zavisnosti od vrednosti težina, što onemogućava mrežu da uči dugoročne zavisnosti[2].

Kako bi rešili ovaj problem, **Hochreiter i Schmidhuber (1997)** uvode arhitekturu **Long Short-Term Memory (LSTM)**. LSTM koristi specijalne *memorijske ćelije* i *gate* mehanizme kako bi obezbedio **konstantan protok gradijenata** kroz duge vremenske intervale[2].

3.1.6 Struktura memorijske ćelije

Prema Hochreiteru i Schmidhuberu [2], svaka memorijska ćelija c_j sastoji se od:

- **Forget gate** f_t – kontroliše koje informacije iz prethodnog stanja treba zaboraviti.
- **Input gate** i_t – odlučuje koje nove informacije ulaze u stanje ćelije.
- **Output gate** o_t – reguliše koje informacije iz stanja ćelije utiču na izlaz.
- **Cell state** C_t – čuva dugoročne informacije kroz sekvencu.

Matematički, osnovne jednačine LSTM-a su:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$
$$h_t = o_t * \tanh(C_t) \tag{6}$$

gde su:

- f_t – *forget gate* koja određuju koje informacije iz C_{t-1} se zadržavaju,
- i_t – *input gate* koja odlučuju koje nove informacije se dodaju u stanje ćelije,
- x_t – ulaz u trenutnom vremenskom koraku,
- h_{t-1} – prethodno skriveno stanje,
- W_f, W_i, W_C, W_o – matrice težina za odgovarajući *gate* i kandidate,
- b_f, b_i, b_C, b_o – vektori pristrasnosti (bias) za odgovarajuća vrata i kandidate,
- $\tilde{C}_t$ – kandidat za novo stanje ćelije (potencijalne vrednosti koje mogu biti dodate u memoriju),
- C_t – stanje ćelije (memorija),
- σ – sigmoidna aktivacija,
- tanh – hiperbolička tangens funkcija.

3.1.7 Prednosti LSTM arhitekture

LSTM poseduje brojne prednosti koje ga čine pogodnim za zadatke sa dugoročnim zavisnostima:

- **Stabilno učenje:** Zahvaljujući konstantnom protoku gradijenata unutar memorijskih ćelija, LSTM uspešno rešava problem dugoročnih zavisnosti i nestajućih gradijenata.
- **Otpornost na šum i fleksibilnost:** Dobro funkcioniše sa šumovitim podacima, kontinuiranim vrednostima i distribuiranim reprezentacijama, bez potrebe za unapred definisanim brojem stanja (za razliku od HMM-a).
- **Generalizacija:** Može da razlikuje udaljene relevantne obrasce u sekvenci, a ne oslanja se samo na kratkoročne signale iz trening skupa.

- **Robusnost:** Dobro radi u širokom opsegu hiperparametara (npr. learning rate, bias vrata), što ga čini stabilnim u praksi.
- **Računska složenost:** LSTM ima vremensku složenost $O(1)$ po težini i vremenskom koraku, što je uporedivo sa klasičnim RNN-ovima. Iako sadrži veći broj parametara, ostaje efikasan za treniranje i inferencu na dugim sekvencama.
- **Široka primena:** Nadmašuje klasične RNN modele u zadacima obrade jezika, prepoznavanja govora, komponovanja muzike i analize vremenskih serija.

Ova arhitektura je ključna za uspeh savremenih NLP zadataka, uključujući i klasifikaciju emocija na osnovu muzičkih tekstova, jer omogućava modelu da prepozna šablone i emotivni kontekst koji se proteže kroz duže delove teksta pesama.

3.1.8 Ograničenja LSTM arhitekture

Iako LSTM arhitektura značajno prevazilazi ograničenja klasičnih RNN modela, ona i dalje ima određene slabosti:

- **Problemi sa specifičnim sekvencama:** Truncated backpropagation verzija LSTM-a teško rešava zadatke poput "delayed XOR", gde se traži računanje XOR operacije nad dva udaljena ulaza.
- **Veća složenost:** Svaki blok memorijskih ćelija sadrži više jedinica (input, forget i output gate), što povećava broj parametara u poređenju sa običnim RNN-ovima.
- **Osetljivost na inicijalizaciju:** U nekim slučajevima, LSTM se ponaša slično kao feedforward mreže koje istovremeno vide celu sekvencu, što može otežati učenje pri velikom broju koraka.
- **Ograničenja u brojanju koraka:** Kao i drugi gradijentni pristupi, nema preciznu sposobnost brojanja vremenskih koraka (npr. razlikovanje između 99 i 100 koraka).

3.1.9 Zaključak o LSTM arhitekturi

Unutrašnja struktura memorijskih ćelija omogućava konstantan protok greške i učenje zavisnosti na velikim vremenskim razmacima. Zbog toga LSTM predstavlja osnovu mnogih modernih NLP i sekvencijalnih modela, uključujući klasifikaciju emocija, obradu govora, komponovanje muzike i predikciju vremenskih serija. I dalje ostaje aktivna oblast istraživanja: efikasniji treninzi, integracija sa Attention i Transformer mehanizmima, adaptacije za specifične domene.

3.2 Transformer modeli

Transformer modeli predstavljaju arhitekturu dubokog učenja specijalno osmišljenu za obradu sekvencijalnih podataka, poput teksta, ali bez korišćenja rekurentnih ili konvolutivnih slojeva. Umesto toga, oni se u potpunosti oslanjaju na *self-attention*, koja omogućava modelu da simultano obradi sve elemente sekvence i da uhvati zavisnosti između reči nezavisno od njihove udaljenosti u tekstu.

Ova arhitektura je prvi put predstavljena u radu *Attention is All You Need* [6], gde je pokazala izuzetnu efikasnost u zadacima mašinskog prevođenja i time označila prekretnicu u oblasti obrade prirodnog jezika (NLP). Od tada, Transformer modeli su postali osnova za većinu savremenih sistema, uključujući BERT, GPT i RoBERTa, i našli primenu u zadacima poput generisanja teksta, sumiranja, analize sentimenta i modeliranja jezika.

3.2.1 Kako Transformer obrađuje tekst

Transformer arhitektura funkcioniše drugačije u odnosu na rekurentne mreže, jer omogućava da se cela sekvenca obrađuje paralelno, bez korak-po-korak obrade. Ključna komponenta je mehanizam *self-attention*, koji omogućava modelu da uvaži odnose između svih reči u ulaznom tekstu, nezavisno od njihove međusobne udaljenosti [6].

Osnovni (uobičajeni) proces rada Transformera može se opisati kroz sledeće faze:

1. **Ulazna reprezentacija:** Svaka reč u ulaznoj sekvenci se konvertuje u vektorsko urezivanje (embedding), a dodatno se primenjuje *poziciono kodiranje* kako bi se modelu obezbedio osećaj redosleda u sekvenci.

2. **Enkoder:** Enkoder paralelno kroz slojeve *self-attention* mehanizama izračunava se odnose između reči, a zatim se rezultati prosleđuju kroz višeslojnu feed-forward mrežu. Svaki sledeći sloj enkodera gradi dublju i apstraktniju reprezentaciju ulaza.

3. **Dekoder:** Dekoder generiše izlaznu sekvencu. On koristi *multi-head pažnju* da se fokusira kako na već generisane izlazne tokene, tako i na reprezentacije ulazne sekvence dobijene iz enkodera.

4. **Izlaz:** Na svakom koraku, dekoder predviđa sledeći token u izlaznoj sekvenci, sve dok se ne dobije kompletan izlaz.

3.2.2 Varijante i unapređenja Transformer arhitekture

Od prvobitnog rada *Attention is All You Need* [6], Transformer arhitektura je doživela brojne nadogradnje. Među značajnijim varijantama su:

- **BERT** (Bidirectional Encoder Representations from Transformers), optimizovan za kontekstualno razumevanje jezika.
- **GPT** (Generative Pretrained Transformer), usmeren na generisanje koherentnog teksta velikih razmera.

- **Transformer-XL**, koji uvodi rekurentni mehanizam i poboljšava modelovanje dužih sekvenci.
- **Reformer**, koji koristi lokalnu osetljivost heširanja za smanjenje računske složenosti.
- **Longformer**, dizajniran za efikasnu obradu dugih dokumenata.

3.2.3 Primena Transformer modela

- **Mašinsko prevođenje**: Transformeri su pokazali superiornost u mašinskom prevođenju, omogućavajući preciznije i skalabilnije sisteme u poređenju sa rekurentnim modelima.
- **Sažimanje teksta**: Modeli poput BART omogućavaju automatsko generisanje sažetaka dužih dokumenat.
- **Generisanje teksta**: Veliki jezički modeli, kao što je GPT, omogućavaju generisanje tečnog i kontekstualno relevantnog teksta
- **Odgovaranje na pitanja**: Modeli BERT i XLNet značajno su unapredili tačnost na zadacima odgovaranja na pitanja na osnovu teksta.
- **Obrada govora i slika**: Varijante Transformer modela našle su primenu i van jezika: za klasifikaciju slika kao i za prepoznavanje govora.

3.2.4 Primena u klasifikaciji emocija

Transformer arhitektura se može efikasno primeniti i u zadacima klasifikacije emocija na osnovu teksta pesama, što predstavlja osnovu za sisteme preporuke muzike zasnovane na lirici. Proces obrade u ovom slučaju funkcioniše na sledeći način:

1. **Ulaz (lyrics):** Tekst pesme se najpre pretvara u numeričke reprezentacije (*embeddings*), kojima se dodaju poziciona kodiranja radi očuvanja redosleda reči.
2. **Enkoder:** Kroz slojeve *self-attention* mehanizma, model uči emocionalne i semantičke obrasce prisutne u lirici. Na primer, fraze poput *"lonely night"* mogu ukazivati na tugu, dok izrazi poput *"bright smile"* sugerišu radost.
3. **Klasifikacioni sloj:** Umesto generisanja nove sekvence (kao kod mašinskog prevođenja), izlaz enkodera prosleđuje se klasifikatoru koji predviđa emocionalnu kategoriju pesme (npr. joy, sadness, anger).
4. **Preporuka muzike:** Na osnovu prepoznate emocije, sistem može da preporuči slične pesme sa istim ili srodnim emocionalnim karakteristikama, da formira plejlistu u skladu sa raspoloženjem korisnika ili da kombinuje liričke i akustičke osobine za preciznije preporuke.

3.2.5 Teorijska osnova i istorijat Transformera

Kao što je već pomenuto osnova Transformer arhitekture počiva na mehanizmu *pažnje* (*attention*), koji omogućava modelu da selektivno fokusira različite delove ulazne sekvence [7].

Prvobitno razvijen za zadatke mašinskog prevođenja, Transformer se brzo proširio na druge oblasti obrade prirodnog jezika, uključujući generisanje teksta, sažimanje i odgovaranje na pitanja. Dalja istraživanja dovela su do stvaranja velikih jezičkih modela, kao što su BERT i GPT, koji su značajno unapredili rezultate na širokom spektru NLP zadataka.

Kasnije su principi Transformera uspešno adaptirani i za druge modalitete, kao što su kompjuterski vid i obrada govora, potvrđujući univerzalnost ove arhitekture.

3.2.6 Ključni koncepti Transformer arhitekture

Self-attention Self-attention mehanizam omogućava svakom tokenu da se poveže sa svim ostalim tokenima u sekvenci. Njegova osnovna formula glasi:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

gde Q, K i V predstavljaju matrice upita, ključeva i vrednosti, a d_k je dimenzija vektora ključeva [6].

Multi-head pažnja Da bi se uhvatile različite reprezentacije iz više podprostora, koristi se multi-head pažnja:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, \ldots, head_h)W^O \tag{8}$$

pri čemu je svaki $head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$, gde su W_i^Q, W_i^K, W_i^V naučljive projektivne matrice [6].

Pozicionalno kodiranje Budući da Transformer nema rekurentnu ili konvolutivnu strukturu, koristi se pozicionalno kodiranje koje uvodi informaciju o redosledu tokena:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right), \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{9}$$

gde *pos* označava poziciju u sekvenci, a i dimenzionalni indeks [6].

Feedforward neuronska mreža Svaka pozicija prolazi kroz potpuno povezanu neuronsku mrežu (FFN) koja se primenjuje nezavisno na svakoj poziciji:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{10}$$

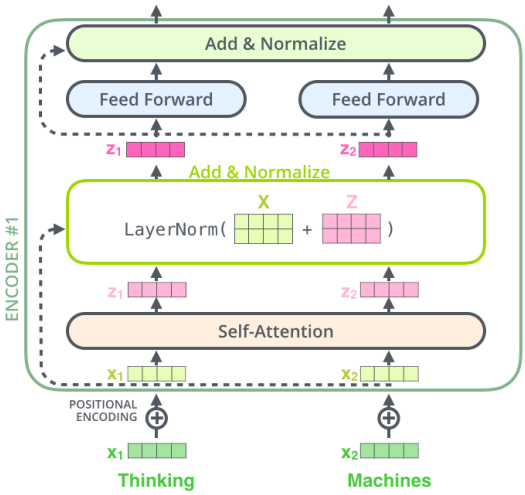
[6].

Normalizacija i rezidualne veze Za stabilnost učenja koriste se rezidualne veze i sloj normalizacije:

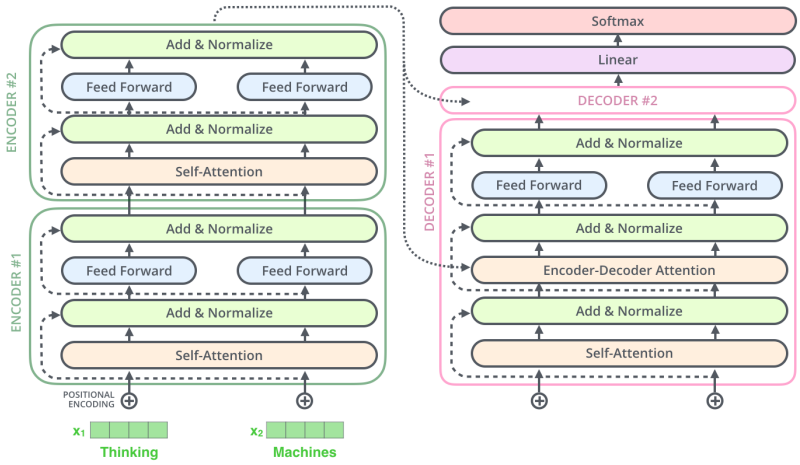
$$\text{LayerNorm}(x + \text{Sublayer}(x)) \tag{11}$$

Rezidualne veze pomažu u očuvanju gradijenata pri dubokom učenju, dok normalizacija standardizuje aktivacije i ubrzava konvergenciju [8].

Vizuelni prikaz arhitekture Na Slici 1 prikazan je primer unutrašnje strukture jednog *encoder* bloka u Transformer arhitekturi. [9].



Slika 1: Unutrašnja struktura jednog Transformer *encoder* bloka.



Slika 2: Kompletna Transformer *encoder–decoder* arhitektura.

Na Slici 2 prikazan je primer kompletne *encoder–decoder* arhitekture. Višeslojni *encoder* (levi deo) generiše reprezentacije ulazne sekvence, koje se prosleđuju u *decoder* (desni deo). *Decoder* blokovi sadrže dodatni sloj *encoder–decoder attention*, koji omogućava modelu da integriše informacije iz izvornog teksta prilikom generisanja ciljne sekvence. [9]

3.2.7 Prednosti Transformer arhitekture

- **Paralelizacija:** Za razliku od RNN modela Transformeri omogućavaju obradu cele sekvence odjednom. Ova osobina značajno ubrzava proces treniranja i omogućava efikasnije korišćenje modernih hardverskih resursa, kao što su GPU i TPU jedinice.
- **Hvatanje dugoročnih zavisnosti:** Mehanizam *self-attention* omogućava da se odnosi između udaljenih tokena u sekvenci modeluju na prirodan i efikasan način.
- **Skalabilnost:** Na njenim osnovama razvijeni su vodeći modeli koji postižu dobre rezultate u raznim sferamaa.
- **Pretreniranje:** Pretrenirani Transformer modeli, poput BERT-a i GPT-a, mogu se dodatno fino prilagoditi za specifične zadatke.

3.2.8 Ograničenja Transformer arhitekture

- **Kvadratna složenost pažnje:** Mehanizam self-attention ima vremensku i memorijsku složenost $O(n^2)$ u odnosu na dužinu sekvence n. Ovo ograničava primenu na veoma dugačke sekvence, gde memorijski trošak postaje prevelik [10].
- **Oslanjanje na veliku količinu podataka:** Efikasnost Transformer modela snažno zavisi od masivnih količina podataka i računske snage.
- **Slabo modelovanje hijerarhijskih struktura:** Iako self-attention dobro hvata semantičke odnose između tokena, Transformeri nemaju ugrađenu sposobnost modelovanja hijerarhijskih struktura jezika (npr. sintaktičkih stabala) [11].
- **Visoki troškovi inferencije:** Veliki pretrenirani modeli, poput GPT ili BERT-large, zahtevaju ogromne resurse ne samo tokom treniranja već i tokom inferencije, što otežava njihovu upotrebu u realnim aplikacijama sa ograničenim resursima.

3.2.9 Zaključak o Transformer arhitekturi

Mehanizam samopažnje i paralelna obrada ulaza omogućavaju efikasno hvatanje dugoročnih zavisnosti i brzu obuku na velikim skupovima podataka. Zbog toga Transformer predstavlja osnovu savremenih NLP modela, uključujući BERT, GPT, T5 i mnoge multimodalne sisteme. I dalje ostaje aktivna oblast istraživanja:

efikasnije skaliranje, smanjenje potrošnje resursa, obrada veoma dugih sekvenci i prilagođavanje specifičnim domenima.

3.3 Rezultati primene modela

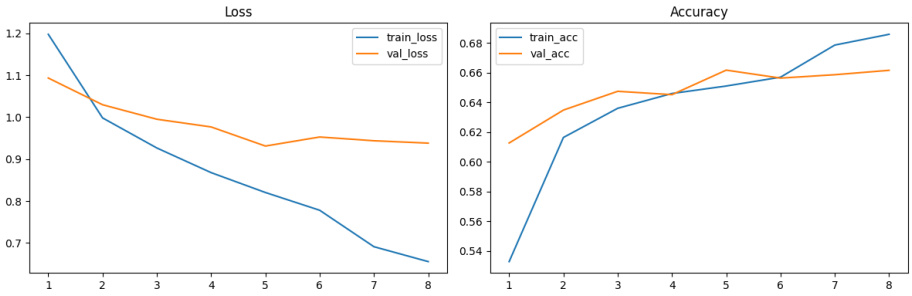
U ovom poglavlju prikazani su rezultati evaluacije modela nad istim test skupom podataka.

3.3.1 LSTM model

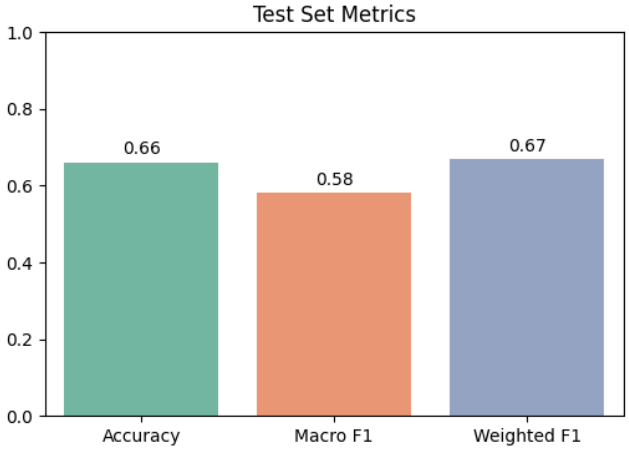
Za treniranje LSTM modela korišćeni su hiperparametri prikazani u Tabeli 1. Oni su podešeni tako da model zadrži dovoljnu izražajnu moć uz kontrolu prenaučenosti pomoću dropout-a.

Parametar	**Vrednost**
Veličina rečnika (max $words$)	20 000
Maksimalna dužina sekvence	128
Dimenzija ugnežđavanja (d_{embed})	100
Broj LSTM jedinica	64
Broj neurona u gustoj sloju	32
Stopa dropout-a	0.3
Veličina mini-batch-a	64
Broj epoha	8
Random seed	42

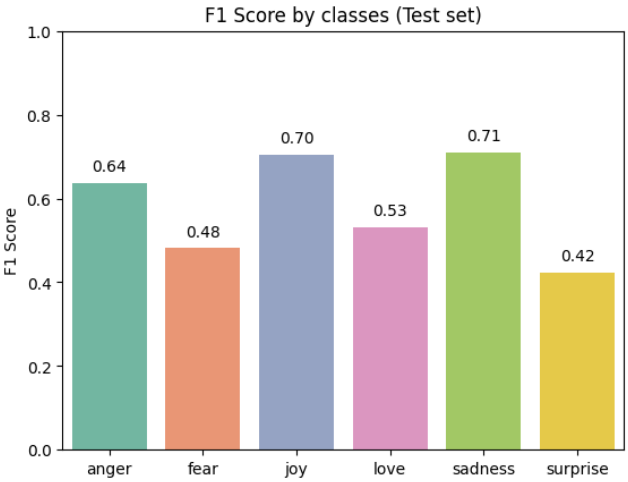
Tabela 1: Hiperparametri korišćenog LSTM modela.



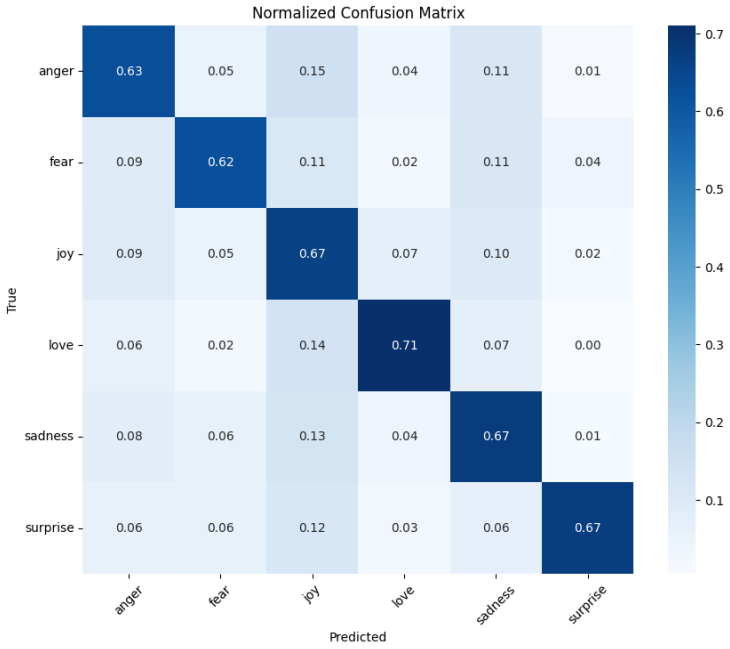
Slika 3: Tok funkcije gubitka i tačnosti tokom epoha.



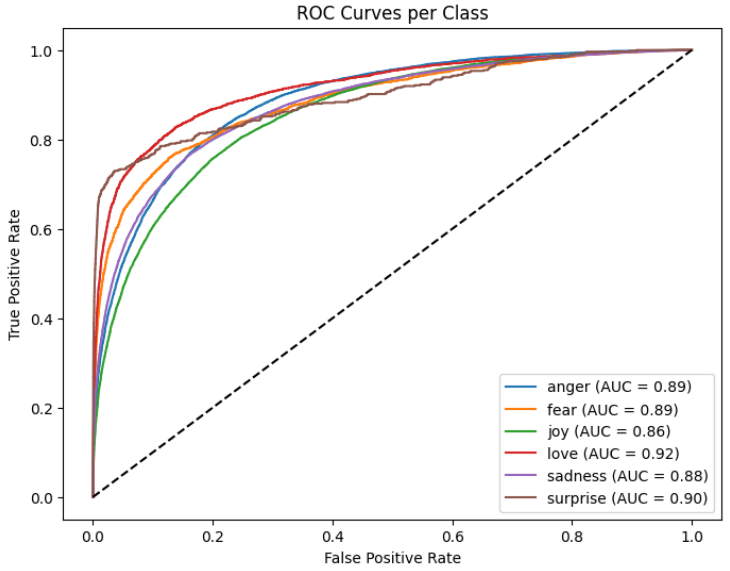
Slika 4: Metrički rezultati modela.



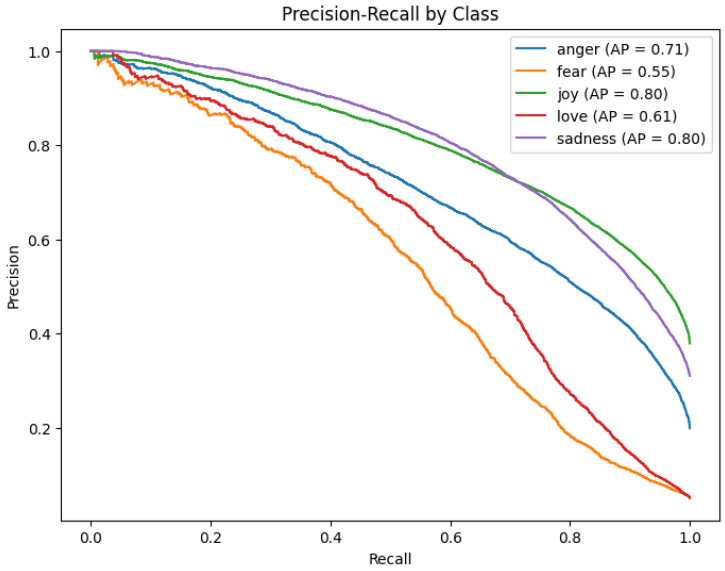
Slika 5: F1 skor po klasama.



Slika 6: Normalizovana matrica konfuzije.



Slika 7: ROC krive po klasama.



Slika 8: Precision–Recall krive po klasama.

Text snippet	**True**	**Pred**	**Prob**
”i told you that i loved you and i meant it then you know i’d never lie to you i do not pretend so do not make up that this was a fake love...”	joy	love	0.798
”merry christmas have a very very merry christmas dream about your heart’s desire christmas eve when you retire santa claus will stop and i know...”	joy	joy	0.999
”this ending is all but an ending he said i do not get why but i might when i am older when we get older when we get older i will get it all...”	anger	sadness	0.727
”i catch a vibe when i am with you lets just lay here until sky blue change every color every hue from stormy grays into sky blue how you...”	joy	joy	0.927
“so many memories and so many miles the road that stretches behind us we have had some laughter and our share of tears but all these moments...”	joy	sadness	0.614

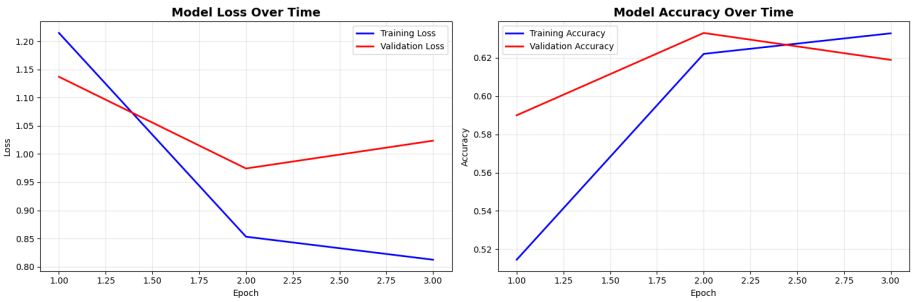
Tabela 2: Primeri predikcija LSTM modela za emocije u pesmama. Tekstovi su skraćeni radi preglednosti.

3.3.2 Transformer model

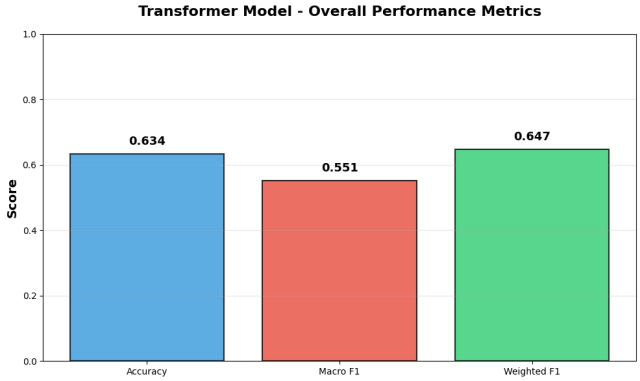
Hiperparametri korišćeni za treniranje Transformer modela prikazani su u Tabeli 3. Ova konfiguracija omogućava balans između izražajne moći i efikasnosti treniranja, uz kontrolu prenaučenosti pomoću dropout-a i ranog zaustavljanja.

Parametar	**Vrednost**
Veličina rečnika (max *tokens*)	10 000
Maksimalna dužina sekvence	384
Dimenzija ugneždavanja (d_{model})	48
Broj attention heads-a (h)	4
Dimenzija feed-forward sloja	192
Broj blokova	1
Stopa dropout-a	0.2
Veličina mini-batch-a	16
Stopa učenja	$2 \cdot 10^{-4}$
Broj epoha	6
Strpljenje (rano zaustavljanje)	1

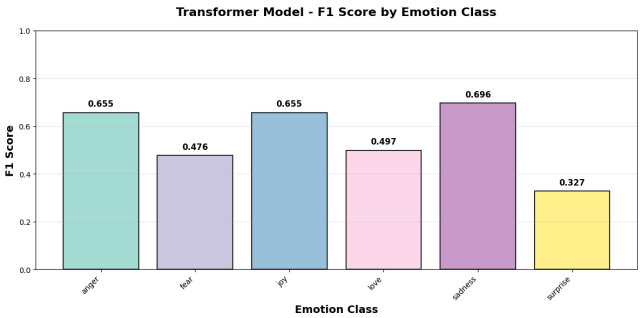
Tabela 3: Hiperparametri korišćenog Transformer modela.



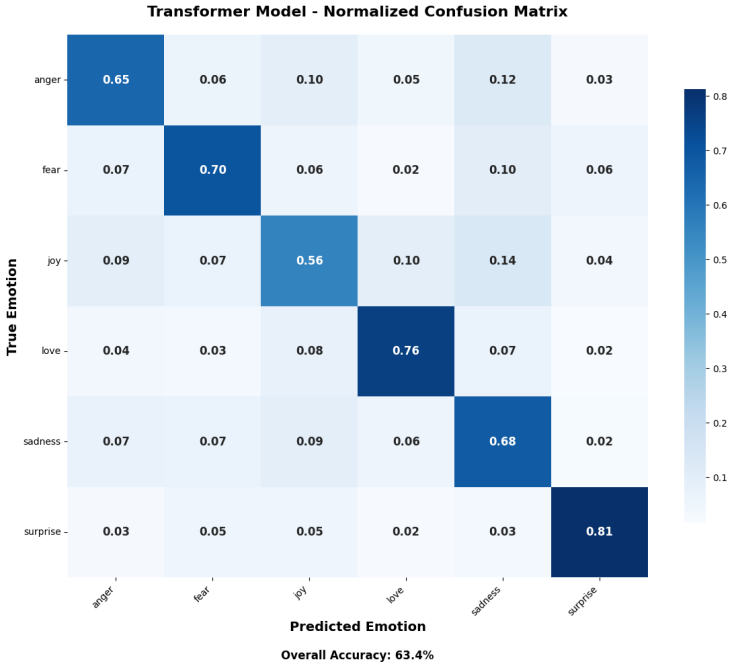
Slika 9: Tok funkcije gubitka i tačnosti tokom epoha.



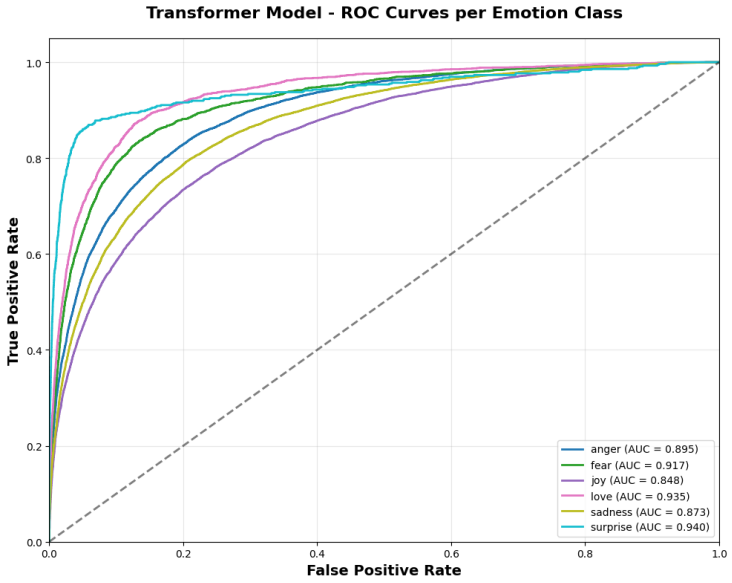
Slika 10: Metrički rezultati modela.



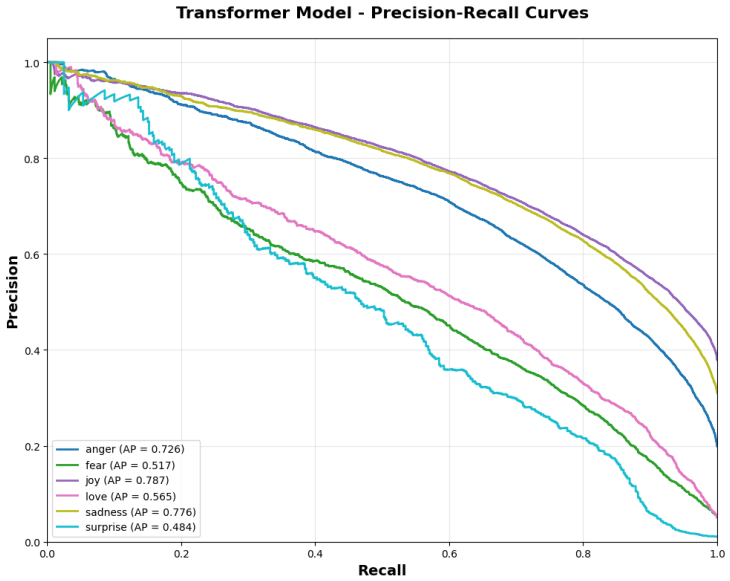
Slika 11: F1 skor po klasama.



Slika 12: Normalizovana matrica konfuzije.



Slika 13: ROC krive po klasama.



Slika 14: Precision–Recall krive po klasama.

Text snippet	**True**	**Pred**	**Prob**
”.she got the red wine in her glass she is ready for the night she is ready for the night she put a painting on the rim of her glass with her lips...”	joy	joy	0.605
”and god said let there be light kings shall bow before you your name will live on when the pyramids are dust moses i have been up to that mountain...”	sadness	sadness	0.798
”got to take the shame from my back it is a hard enough life for the two of us we both know we are superstars i want to take the blame but i...”	sadness	fear	0.348
”merry christmas have a very very merry christmas dream about your heart’s desire christmas eve when you retire santa claus will stop and i know...”	joy	joy	0.971
”i catch a vibe when i am with you lets just lay here until sky blue change every color every hue from stormy grays into sky blue how you...”	joy	sadness	0.599

Tabela 4: Primeri predikcija Transformer modela za emocije u pesmama. Tekstovi su skraćeni radi preglednosti.

4 Zaključak

U ovom radu uporedili smo performanse LSTM i Transformer modela u zadatku klasifikacije emocija u muzičkim tekstovima. Rezultati su pokazali da oba modela ostvaruju uporedive performanse, pri čemu je LSTM ostvario blago bolje rezultate u pogledu tačnosti i F1 skora.

To potvrđuje da, uprkos dominaciji Transformer arhitektura u savremenim NLP zadacima, u slučaju kada se ne koriste unapred obučeni tokenizatori sa naprednijim NLP sposobnostima (da budu svesni konteksta ili podreči), LSTM modeli i dalje mogu biti konkurentni, posebno u specijalizovanim domenima i kada se raspolaže ograničenim količinama podataka.

Budući rad može uključivati kombinovanje LSTM i Transformer pristupa, dodavanje unapred obučenih tokenizatora sa naprednijim NLP sposobnostima ili integraciju mehanizma pažnje radi daljeg poboljšanja performansi.

Literatura

[1] https://www.kaggle.com/datasets/devdope/900k-spotify

[2] S. Hochreiter, J. Schmidhuber. *Long Short-Term Memory*. Neural Computation, vol. 9, no. 8, pp. 1735–1780, MIT Press, 1997.

[3] M. Krichen, A. Mihoub. *Long Short-Term Memory Networks: A Comprehensive Survey*. Department of Software Engineering, Albaha University; ReDCAD Research Laboratory, Sfax University; Department of Management Information Systems, Qassim University, 2025.

[4] J. Kumar, R. Goomer, A. K. Singh. *Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model for Cloud Datacenters*. 6th International Conference on Smart Computing and Communications (ICSCC), Kurukshetra, India, Elsevier B.V., 2018.

[5] H. Sak, A. Senior, F. Beaufays. *Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling*. Proc. Interspeech, Google Inc., USA, 2014.

[6] Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N and Kaiser, Łukasz and Polosukhin, Illia. *Attention Is All You Need*. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017.

[7] Bahdanau, Dzmitry and Cho, Kyunghyun and Bengio, Yoshua. *Neural Machine Translation by Jointly Learning to Align and Translate*. ICLR, 2015.

[8] Ba, Jimmy Lei and Kiros, Jamie Ryan and Hinton, Geoffrey. *Layer Normalization*. arXiv preprint arXiv:1607.06450, 2016

[9] https://jalammar.github.io/illustrated-transformer/

[10] Child, Rewon and Gray, Scott and Radford, Alec and Sutskever, Ilya. *Generating Long Sequences with Sparse Transformers*. arXiv preprint arXiv:1904.10509, 2019.

[11] Linzen, Tal and Baroni, Marco. *Can RNNs learn hierarchical generalization?* Transactions of the Association for Computational Linguistics, volume 7, 2019.

[12] https://github.com/jjulijana/music_lyrics_classification