Generiranje tekstova pjesama pomoću RNN

Tomislav Krog
Fakultet elektrotehnike i računarstva
Sveučilišta u Zagrebu
Zagreb, Hrvatska
tomislav.krog@fer.hr

Matija Sever
Fakultet elektrotehnike i računarstva
Sveučilišta u Zagrebu
Zagreb, Hrvatska
matija.sever@fer.hr

Jurica Matošić
Fakultet elektrotehnike i računarstva
Sveučilišta u Zagrebu)
Zagreb, Hrvatska
jurica.matosic@fer.hr

Lara Kokeza
Fakultet elektrotehnike i računarstva
Sveučilišta u Zagrebu
Zagreb, Hrvatska
lara.kokeza@fer.hr

Luka Maros
Fakultet elektrotehnike i računarstva
Sveučilišta u Zagrebu
Zagreb, Hrvatska
luka.maros@fer.hr

Rej Šafranko
Fakultet elektrotehnike i računarstva
Sveučilišta u Zagrebu
Zagreb, Hrvatska
rej.safranko@fer.hr

Sažetak: ***U području generiranja sadržaja, generiranje tekstova pjesama je izazovan pothvat. Ovaj projekt istražuje i uspoređuje modele dubokog učenja koji se koriste za generiranje tekstova pjesama: povratni model BiLSTM i veliki jezični modeli GPT-2 te Tiny Llama. Cilj nam je bio razviti generički generator stihova i generator koji oponaša određenog izvođača. Naš projekt procjenjuje kreativne sposobnosti i lirsku koherentnost ovih modela za rješavanje naših zadataka. Eksperimentiranje se provodi na opsežnom skupu podataka postojećih tekstova pjesama modeliranih iz dva izvora podataka, a modeli se ocjenjuju na temelju njihove sposobnosti da proizvedu kontekstualno relevantne i umjetnički koherentne tekstove. Uspoređujući generirane tekstove s tekstovima iz stvarnog svijeta, ovaj projekt pridonosi raspravi o sjecištu dubokog učenja i kreativnog izražavanja u polju pisanja pjesama.***

Ključne riječi: generiranje, lirika, duboko učenje, povratni model, veliki jezični model, pjesme

I. UVOD

U današnje vrijeme, uporaba modela dubokog učenja u umjetničkom stvaralaštvu označava uzbudljiv napredak na raskrižju tehnologije i kreativnog izražavanja. Iskorištavajući mogućnosti povratnih modela i velikih jezičnih modela, ovaj projekt zadire u proces stvaranja uvjerljivih stihova pomoću umjetne inteligencije. Pojava dubokog učenja promijenila je krajolik obrade prirodnog jezika, pružajući alate poput povratnih modela koji se ističu u učenju sekvencijskih ovisnosti. Dvosmjerna mreža dugog kratkoročnog pamćenja (BiLSTM) je vrsta povratnog modela koji nudi jedinstvenu sposobnost razumijevanja uzoraka u sekvencijskim podacima. To ih čini posebno prikladnima za zamršenu strukturu tekstova pjesama. S druge strane, veliki jezični modeli i njihove verzije s manje parametara, kao što je arhitektura Tiny Llama, stavljaju u prvi plan sposobnost razumijevanja suptilnosti jezika na široj razini. Ovi modeli uče iz golemih količina tekstualnih podataka, što im omogućuje da proizvedu tekstove koji ne samo da oponašaju utvrđene obrasce, već također pokazuju smisao za kreativnost i inovaciju. Naš projekt istražuje i uspoređuje učinkovitost ovih modela dubokog učenja u domeni generiranja stihova pjesama. Nastojimo razviti generički generator stihova te generator stihova koji oponaša određenog izvođača.

II. PREGLED LITERATURE

Generiranje tekstova pjesama, potaknuto napretkom dubokog učenja i obrade prirodnog jezika, postalo je istaknuto područje istraživanja prethodnih godina. Istraživanje inovativnih tehnika i modela za generiranje kreativnih i kontekstualno koherentnih tekstova privuklo je značajnu pozornost unutar akademske zajednice. Ovaj pregled literature ima za cilj pružiti uvid u razvoj ključnih tema i metodologija od ranih pokušaja sekvencijskog modeliranja i generiranja do najnovijih praksi današnjeg vremena.

A. Povratne neuronske mreže (RNN) i LSTM modeli

Rana istraživanja o stvaranju tekstova pjesama često su koristila mreže povratnih neuronskih mreža (RNN) i dugotrajnog kratkoročnog pamćenja (LSTM) zbog njihove sposobnosti u učenju sekvencijskih ovisnosti. Istraživanje Ecka i Schmidhubera (2002.) [1] koristilo je LSTM mreže za generiranje glazbe, postavljajući temelj za kasniji rad na generiranju stihova.

B. Dvosmjerni modeli za poboljšanu koherenciju

Dvosmjerni LSTM modeli pojavili su se kao odgovor na potrebu za poboljšanim razumijevanjem konteksta. Cho i suradnici (2014.) [2] demonstrirali su učinkovitost dvosmjernih modela u učenju dvosmjernih ovisnosti što pridonosi poboljšanoj koherentnosti i protoku u generiranim tekstovima.

C. Veliki jezični modeli i kreativnost

Pojava velikih jezičnih modela, kao što je GPT-3, otvorila je novu eru stvaranja lirike. Studije poput Browna (2020.) [3] pokazale su kreativni potencijal takvih modela u rješavanju raznih zadataka obrade prirodnog jezika. To uključuje i generiranje tekstove koji ne samo da oponašaju postojeće obrasce, već uvode i nove i maštovite izraze.

D. Generiranje stihova rap pjesama

"DopeLearning: A Computational Approach to Rap Lyrics Generation" (Malmi i suradnici, 2015.) [4] predstavlja novu metodu za generiranje rap stihova korištenjem modela predviđanja temeljenog na RankSVM algoritmu i dubokoj neuronskoj mreži. Model pokazuje 17% točnosti u identificiranju sljedećeg stiha među 299 slučajno izabranih stihova, značajno nadmašujući slučajni odabir. Također model pokazuje poboljšanje u gustoći rime za 21% u usporedbi s ljudskim reperima. Rad predstavlja primjer uspješne integracije računalnih tehnika i kreativnosti u polju generiranja rap pjesama.

III. OPIS RJEŠENJA

A. Ulazni skup podataka

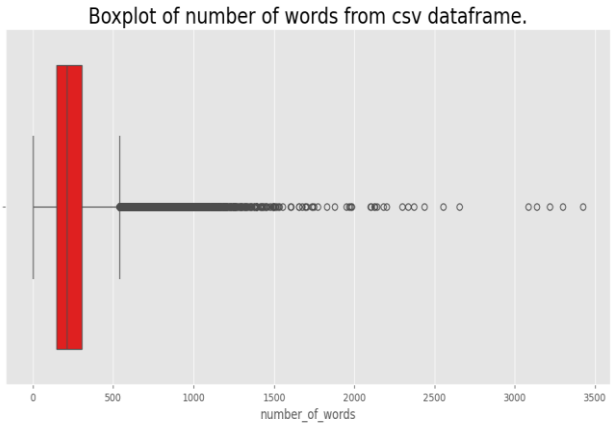
Ulazni skup podataka je modeliran iz dva izvora podataka. Jedan izvor je CSV datoteka gdje svaki redak odgovara jednoj pjesmi, dok je drugi izvor skup tekstualnih datoteka gdje svaka datoteka odgovara jednom umjetniku. Unutar tekstualne datoteke su svi stihovi tog umjetnika. Ne postoje separacijski znakovi koji bi odjelili stihove po pjesmama nego su svi stihovi odijeljeni znakom nove linije.

Prvi zadatak je bio odabrati značajke CSV datoteke. Odbacili smo pjesme koje nisu na engleskom jeziku i zadržali samo značajke koje odgovaraju umjetnicima i pjesmama. Standardizirali smo nazive umjetnika i u CSV datoteci i u nazivima tekstualnih datoteka kako bi lakše nadopunili CSV datoteku s umjetnicima (i njihovim odgovarajućim stihovima) koji nisu bili prisutni u njoj.

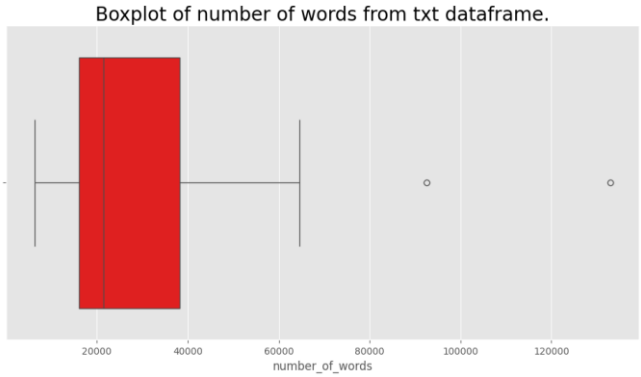
Idući zadatak vezan uz pripremu ulaznog skupa podataka je bio čišćenje i obrada nadopunjene CSV datoteke. Standardizirali smo stihove zamjenom višestrukih praznina s jednom prazninom, ali smo ostavili znak nove linije s ciljem da naučimo model odvajati stihove. Izbacili smo sve posebne znakove osim ".", "?", i "!" pošto interpunkcijski znakovi nose strukturnu i kontekstualnu informaciju u stihovima. Cilj je bio zadržati čim više takvih informacija kako bi ispitali može li ih model naučiti ispravno generirati u stihovima.

Pošto su stihovi kojima smo nadopunili CSV datoteku zapravo svi stihovi nekog umjetnika, postoji bipolaritet u broju riječi stihova koji su iz CSV izvora i onih koji su iz tekstualnih datoteka.

	Originalni izvori podataka	
Broj riječi	CSV	TXT
Srednja vrijednost	250.2292	29948.4444
Standardna devijacija	159.2884	23189.7702
Minimum	1.0000	6343.0000
Maximum	3422.0000	132909.0000



Slika 1 - Boxplot broja riječi stihova iz CSV datoteke



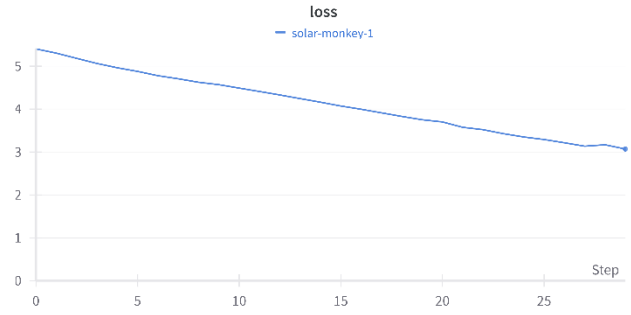
Slika 2 - Boxplot broja riječi stihova iz tekstualnih datoteka

Ovaj problem smo riješili tako što smo razbili stihove dodane iz tekstuanlih datoteka na manje dijelove u veličini od 250 riječi. Taj broj riječi odgovara srednjoj vrijednosti broja riječi pjesama iz CSV datoteke. Ovim postupkom smo željeli održati konzistentnost sa statistikom broja riječi pjesama iz CSV datoteke. Smatramo da je to nužno jer je velik omjer originalnih CSV redaka (pjesama) naspram dodanih redaka iz tekstualnih datoteka – 191814: 45. Nakon uravnotežavanja broja riječi po stihovima, navedeni omjer postaje 191814: 7001. Također, maknuli smo redak koji ne sadrži više od jedne riječi.

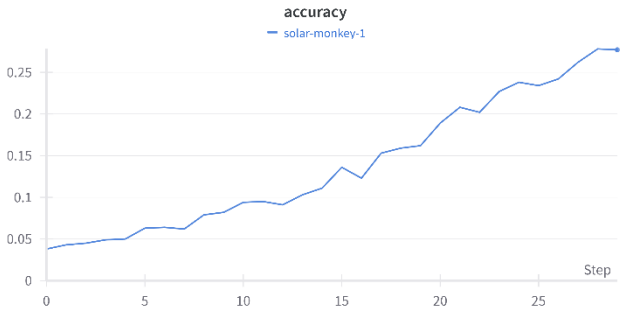
B. Generator generičkih stihova

1) BiLSTM

U ovom projektu razvili smo BiLSTM (Bidirectional Long Short-Term Memory) model za zadatak generiranja tekstova pjesama. Ulazni skup podataka je tokeniziran Keras modelom tokenizatora koji olakšava konverziju pjesama u numeričke nizove. Kako bi osigurali ponovljivost, model tokenizacije smo serijalizirali je i spremili za buduću upotrebu. Tokenizirane sekvence smo dalje transformirali u sekvence n-grama kako bi uhvatili kontekstualne ovisnosti unutar teksta. Maksimalnu duljinu sekvence, ključni parametar za učenje modela, odredili smo analizom duljina generiranih sekvenci. Kako bi optimizirali upotrebu memorije, sekvence smo dopunili u manjim skupovima. Naposlijetku smo stvorili značajke (X) i oznake (y) za model. Oznaka je zadnja riječ sekvence koju želimo predvidjeti dok su značajke sve prijašnje riječi sekvence. Oznake smo kodirali kako bi predstavili kategoričku prirodu zadatka. Rezultirajući skup podataka, koji sadrži značajke i oznake, organiziran je u rječnik za jednostavniji pristup. BiLSTM model konstruirali smo korištenjem Kerasa, a sastoji se od sloja za ugradnju, dvosmjernog LSTM sloja s 250 jedinica, dropout sloja za regularizaciju i potpuno povezanog sloja sa softmax aktivacijom za predviđanje sljedeće riječi u nizu. Model je sastavljen s kategoričkim unakrsnim gubitkom entropije i Adamovim optimizatorom. Proces treninga nadziran je pomoću EarlyStoppinga kako bi se spriječila prenaučenost modela, a WandB (Weights and Biases) integriran je za sveobuhvatno praćenje napretka učenja modela. Model je prošao obuku kroz 30 epoha.



Slika 3 - gubitak modela u ovisnosti o broju epoha



Slika 4 - točnost modela u ovisnosti o broju epoha

2) Varijacijski autoenkoder

Cilj ovog dijela projekta je generiranje novih tekstova pjesama korištenjem Varijacijskog Autoenkodera (VAE).

Odabir skupa podataka: Koristio se skup podataka koji sadrži tekstove pjesama. Podaci su učitani iz CSV datoteke, a zatim je nasumično odabrano 130,000 uzoraka. Tokenizacija teksta: Tekstovi su tokenizirani pomoću Keras Tokenizera. Tekst je pretvoren u mala slova, a Tokenizer je prilagođen tekstovima kako bi se stvorio vokabular. Tokenizirane sekvence su zatim proširene kako bi se osigurala jednaka duljina tijekom treniranja. Analiza podataka: Histogramom koji prikazuje distribuciju duljina sekvenci u tokeniziranim tekstovima kao najčešća duljina sekvence identificirana je duljina 200 te su sve sekvence prilagođene na tu duljinu.

Arhitektura Modela:

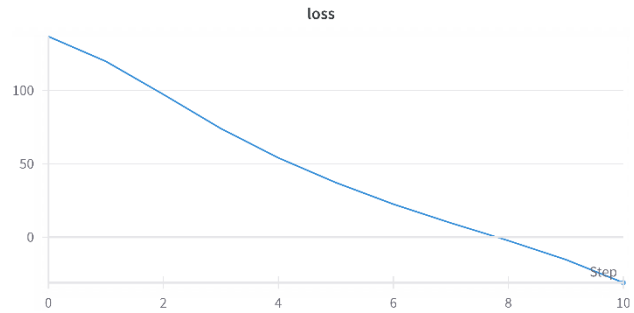
Encoder: Encoder obrađuje proširene sekvence kroz sloj ugniježđenja, zatim sloj ravnanja i sloj gustoće s ReLU aktivacijom. Latentni prostor: Srednja vrijednost i log varijanca latentnog prostora predviđaju se pomoću dvaju odvojenih slojeva gustoće. Trik reparametrizacije: VAE koristi trik reparametrizacije za uzorkovanje iz latentnog prostora, poboljšavajući stabilnost treniranja. Decoder: Decoder se sastoji od slojeva gustoće s ReLU aktivacijom i završnog sloja gustoće s sigmoidalnom aktivacijom za rekonstrukciju izvorne sekvence. Prilagođeni varijacijski sloj: Dodan je prilagođeni sloj za izračun gubitka VAE-a, kombinirajući gubitak rekonstrukcije i KL divergencije.

Treniranje modela:

Kompilacija modela: VAE model je kompiliran pomoću optimizatora Adam s početnom stopom učenja od 0.00001. Funkcija gubitka specificirana je kao prilagođena funkcija gubitka definirana u već spomenutom prilagođenom varijacijskom sloju.

Parametri treniranja: Model je treniran tijekom 10 epoha s grupnom veličinom od 512. Primijenjeno je rano zaustavljanje kako bi se spriječilo prenaučavanje, a napredak treniranja praćen je pomoću TensorBoarda. Gubitci Treniranja i validacije: Gubitci treniranja i validacije prikazani su tijekom epoha radi vizualizacije procesa učenja. Trenirani VAE model spremljen je za buduću upotrebu.

Praćenje Projekta WandB integracijom: Projekt je integriran s Weights & Biases (WandB) za praćenje eksperimenata. WandB se koristi za zapisivanje metrika treniranja i vizualizaciju eksperimenta.



Slika 5 - gubitak modela u ovisnosti o broju epoha

C. Generator stihova koji oponaša umjetnika

*1) **GPT-2***

U sklopu ovog projekta, razvijen je sofisticirani sustav za generiranje tekstova pjesama koristeći model GPT-2. Naš pristup uključuje prilagodbu i finu obradu prethodno treniranog GPT-2 modela na specifičnom skupu podataka, koji se sastoji isključivo od pjesama odabranog izvođača. Ova strategija je odabrana jer je ključno da, ako želimo uspješno imitirati stil i izričaj pojedinog izvođača, model bude fino podešen (*engl. fine tuned*) na tekstovima samo tog izvođača. Takav pristup omogućava modelu da uhvati i nauči jedinstvene jezične obrasce, fraziranje i emotivni izraz koji karakteriziraju specifičnog umjetnika. Ako bi se model trenirao na širokom spektru tekstova različitih izvođača, postojao bi značajan rizik od gubitka tih jedinstvenih stilskih karakteristika, što bi rezultiralo generiranjem tekstova koji možda ne bi adekvatno reflektirali specifičan umjetnički izričaj odabranog izvođača.

Pretprocesuiranje tekstova pjesama ključni je korak u cijelom procesu. Korištena je funkcija *preprocess_lyrics* za čišćenje teksta od nepotrebnih elemenata kao što su zagrade, specijalni simboli i neželjeni znakovi interpunkcije. Osim toga, tekstovi su transformirani kako bi se uklonile sve vrste nepotrebnog šuma, poput nepotrebnih riječi ili fraza koje nisu relevantne za kontekst pjesme. Ovaj korak osigurava da svaki ulaz u model bude čist i precizno usklađen sa stilom i tematikom izvođača. Poboljšanjem kvalitete ulaznih podataka, model je mogao s većom preciznošću generirati tekstove koji vjerno odražavaju umjetnički izričaj i emocije pjesama.

Tokenizer, koji je dio *transformers* biblioteke, korišten je za konverziju teksta u niz numeričkih tokena koji služe kao ulaz u model. GPT-2, poznat po svojoj sposobnosti da hvata duboke kontekstualne ovisnosti u jeziku, idealan je za stvaranje koherentnih i relevantnih tekstova pjesama.

Tijekom procesa treniranja i finog podešavanja modela, korišten je AdamW optimizator, s postupkom postepenog povećavanja i smanjivanja stope učenja. Model je treniran kroz 15 epoha uz pažljivo praćenje prekomjernog učenja korištenjem metoda poput ranog zaustavljanja (*engl. early stopping).* Integracija s platformom WandB (Weights and Biases) omogućila je detaljno praćenje i vizualizaciju napretka tijekom procesa učenja, uključujući praćenje gubitaka *(engl. loss*) u svakoj epohi.

Rezultat našeg rada je GPT-2 model sposoban za efikasno generiranje autentičnih i stilski dosljednih tekstova pjesama. Ovaj model demonstrira zadovoljavajuće mogućnosti primjene metoda dubokog učenja u zadacima poput generiranja umjetničkih tekstova.

Slika 6 gubitak modela GPT-2 u ovisnosti o broju epoha

2) *Tiny Llama*

Paralelno s GPT-2 modelom, razvijen je i generator stihova koristeći Tiny Llama model. Ovaj generator koristi naprednu tehnologiju preoblikovanja modela (*engl. model reshaping technology*) kako bi se prilagodio specifičnim potrebama generiranja tekstova pjesama. Naš se generator temelji na sofisticiranom pristupu koji uključuje korištenje PEFT (Parameter Efficient Fine-Tuning) tehnike. Ova tehnika omogućuje fino podešavanje modela s većom efikasnošću, koristeći manje resursa bez gubitka performansi.

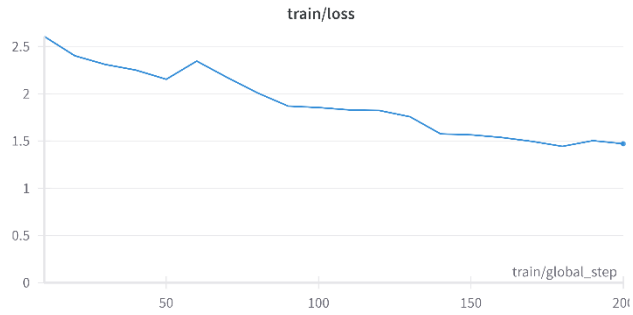
Slično kao i s GPT-2 modelom, Tiny Llama prilagođen je specifičnom skupu podataka, sastavljenom od pjesama odabranog izvođača. Razlog za ovakav pristup je identičan - očuvanje jedinstvenog stila i izričaja izvođača.

Model je konstruiran s integracijom različitih tehnologija, uključujući PEFT i BitsAndBytes, što omogućava efikasnije treniranje modela i optimizaciju upotrebe memorije. Korištenje trl (Transfer and Reinforcement Learning) trenera, s HuggingFace integracijom, dodatno poboljšava proces treniranja i omogućava lakše dijeljenje rezultata.

Tiny Llama model treniran je koristeći prilagođene postavke, kao što su manja veličina serije, akumulacija gradijenta, i sofisticirani algoritam optimizacije. Za kontrolu procesa učenja, korišten je pristup koji uključuje kosinusno podešavanje stope učenja te rano zaustavljanje kako bi se izbjeglo prenaučavanje.

Integracija s platformom WandB (Weights and Biases) omogućila je detaljno praćenje i vizualizaciju napretka tijekom procesa učenja, uključujući praćenje gubitka u svakom koraku algoritma.

Rezultat rada s Tiny Llama modelom visoko je učinkovit sustav sposoban za generiranje autentičnih i stilski dosljednih tekstova pjesama, uz značajne uštede u resursima i vremenu potrebnom za treniranje.



Slika 7 - gubitak modela Tiny Llama u ovisnosti o broju koraka

D. Evaluacija modela

Evaluacija modela za generiranje stihova provedena je korištenjem četiri različita pristupa: bidirekcionalni LSTM, varijacijski autoencoder, GPT-2 i TinyLlama. Postupak evaluacije bio je sljedeći: svakom od ovih modela dano je nekoliko početnih riječi iz stvarnih pjesama, na temelju kojih su modeli generirali nastavke. Za svaki model generirano je ukupno deset različitih stihova.

Ovi generirani stihovi potom su podvrgnuti ocjenjivanju na osnovi njihove realističnosti. Ocjenjivanje su provodila četiri člana našeg tima, a dodatnu evaluaciju pružio je i jezični model GPT-4. Kriterij za ocjenjivanje bio je jednostavan: ako stihovi zvuče realistično, modelu je dodijeljena ocjena 1; u suprotnom, ocjena 0. Za procjenu učinkovitosti svakog modela koristili smo metriku konačne ocjene. Ova metrika izračunata je zbrajanjem svih jedinica (ocjena 1) koje je svaki model dobio, a zatim se taj zbroj podijelio s ukupnim brojem ocjena (koji je u ovom slučaju 50, jer je svaki od 4 člana tima (plus GPT-4) dao ocjenu za 10 stihova svakog modela). Tako dobiveni rezultat odražava postotak točnosti, odnosno efikasnost svakog modela u generiranju realističnih stihova.

Ova metoda omogućila je precizno uspoređivanje performansi svakog modela, dajući nam jasan uvid u njihove sposobnosti i ograničenja u stvaranju uvjerljivih i koherentnih stihova. Rezultati ove evaluacije bit će ključni za daljnji razvoj i usavršavanje modela za generiranje stihova.

Modeli BiLSTM i VAE nisu uspijeli ostvariti pozitivne rezultate u ovoj evaluaciji, s točnošću (accuracy) od 0 %. To znači da su svi ocjenjivači svih deset stihova koje su generirali ovi modeli ocijenili kao nerealistične. Nasuprot tome, GPT-2 model ostvario je znatno bolje rezultate s accuracy od 72 % (0.72), što ukazuje na visoku razinu realističnosti većine stihova koje je generirao. Najbolji rezultat pokazao je TinyLlama model, s najvišim postotkom točnosti od 74 % (0.74), čime se ističe kao najuspješniji model u generiranju uvjerljivih stihova u ovom eksperimentu. Svi ovi rezultati prikazani su u Tablica 1.**Pogreška! Izvor reference nije pronađen.**

	BiLSTM	VAE	GPT-2	Tiny LLama
Konačna ocjena	0.00	0.00	0.72	**0.74**
GPT-4 ocjena	0.00	**0.00**	**0.80**	**0.80**

Tablica 1 performanse modela

INPUT:

At least I can say that I've tried
To tell you I'm sorry
For breaking your heart
But it don't matter, it clearly

OUTPUT:

Doesn't change my mind
You are still there
Whenever we talk now
We always end up at our own ends
So let us both know where we stand

Slika 8 Primjer generiranja stihova modela Tiny Llama

Set 2

1. 1 (Realistic: Reflects on personal experiences and life choices)
2. 1 (Realistic: Describes romantic anticipation and vulnerability)
3. 1 (Realistic: Expresses desire for love and connection)
4. 0 (Not very realistic: Exaggerated and somewhat humorous portrayal of family dynamics)
5. 0 (Not very realistic: Describes extreme and unrealistic actions)
6. 1 (Realistic: Asserts individuality and personal legacy)
7. 1 (Realistic: Describes remorse and hope for reconciliation)
8. 1 (Realistic: Expresses longing and readiness for renewed love)
9. 1 (Realistic: Apologetic tone, acknowledging past mistakes in a relationship)
10. 1 (Realistic: Describes a longing for reconnection and time running out)

Slika 9 GPT-4 ocjenjivanje Tiny Llama generiranih stihova (podebljano: ocjena stihova sa slike 6)

REFERENCE

[1] J. Eck and J. Schmidhuber, "LSTM Network for Generating Music," Neural Networks Research, 2002.

[2] K. Cho et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," arXiv:1406.1078, 2014.

[3] T. B. Brown et al., "Language Models are Few-Shot Learners," arXiv:2005.14165, 2020.

[4] E. Malmi et al., "DopeLearning: A Computational Approach to Rap Lyrics Generation," in Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1959–1968