UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN INFORMACIJSKE TEHNOLOGIJE

ZAKLJUČNA NALOGA

GENERIRANJE POSVETIL S POMOČJO NEVRONSKIH MREŽ

GAŠPER SEVER

ZAKLJUČNA NALOGA

SEVER

2023

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

Generiranje posvetil s pomočjo nevronskih mrež

(Generating dedications using neural networks)

Ime in priimek: Gašper Sever
Študijski program: Računalništvo in informatika
Mentor: izr. prof. dr. Jernej Vičič

Koper, januar 2023

Ključna dokumentacijska informacija

Ime in PRIIMEK: Gašper SEVER

Naslov zaključne naloge: Generiranje posvetil s pomočjo nevronskih mrež

Kraj: Koper

Leto: 2023

Število listov: 41 Število slik: 16 Število tabel: 6

Število prilog: 1 Število strani prilog: 3 Število referenc: 26

Mentor: izr. prof. dr. Jernej Vičič

Ključne besede: posvetilo, akrostih, izdelava rime, nevronska mreža, generiranje besedila, semantika

Izvleček:
V diplomski nalogi je predstavljena izdelava generatorja posvetil. V prvem delu je predstavljen postopek, kako generator dobi podatke iz učnih korpusov, kateri od njih so izbrani, kako se na podatkih uči in kako iz njih naredi model, s pomočjo katerega si zapiše naučene podatke. Nato je predstavljena tudi druga nevronska mreža, ki je uporabljena za učenje rim. V drugem delu je predstavljena implementacija generiranja posvetil v angleškem jeziku. Sledi primerjava učinkovitosti rezultata, z učno množico, kjer je zbrana množica francoskih posvetil z angleškimi posvetili. V zaključku so podane možne izboljšave in predlogi za nadaljnje delo.

Key document information

Name and SURNAME: Gašper SEVER

Title of the final project paper: Generating dedications using neural networks

Place: Koper

Year: 2023

Number of pages: 41 Number of figures: 16 Number of tables: 6

Number of appendices: 1 Number of appendix pages: 3 Number of references: 26

Mentor: Assoc. Prof. Jernej Vičič, PhD

Keywords: dedications, acrostic, generating rhyme model, neural network, generating text, semantics

Abstract:
The thesis presents the creation of an acrostic poem generator. The first part presents the process of how the generator obtains data from the learning corpus, which of them were chosen, how it learns from the data, and how it makes a model out of given data, with the help of which it remembers the learned data. Then another neural network is presented, which is used for learning rhymes. In the second part, the implementation of the dedication generator in the English language is presented. Following it, is a comparison of the effectiveness of the English results, with the learning set, where a set of French dedications are generated. Possible improvements and suggestions for further work are given in the conclusion.

Zahvala

Rad bi se zahvalil mentorju, izr. prof. dr. Jerneju Vičiču, za vso podporo, strokovno pomoč in usmeritve pri zaključnem delu in v času izobraževanja. Poleg tega bi se rad zahvalil svoji družini za vso izkazano podporo. Posebna zahvala gre vsem kolegom, s katerimi smo se skupaj učili in nadgrajevali znanje tekom študija.

Kazalo vsebine

Kazalo tabel

Kazalo slik

Seznam kratic

AI	ang. Artificial intelligence (slov. umetna inteligenca)
BPTT	ang. Backpropagation Through Time (slov. razmnoževanje skozi čas)
GRU	ang. Gated recurrent unit (slov. nevronska celica z vrati)
LSTM	ang. Long short-term memory (slov. nevronska celica z dolgim kratkotrajnim spominom)
ML	ang. Machine learning (slov. strojno učenje)
NLP	ang. Natural Language Processing (slov. enota za procesiranje naravnega jezika)
NLTK	ang. Natural Language Toolkit, python knjižnica
NMF	ang. Nonnegative matrix factorization (slov. množenje nenegativnih matrik)
RNN	ang. Recurrent neural network (slov. ponavljajoča nevronska mreža)
URL	ang. Uniform Resource Locator (slov. enolični krajevnik vira)

1 Uvod

Posvetila so ena izmed redkih vrst besedil, kjer se nam bo umetna inteligenca najbolj približala, saj ne potrebujejo visoke stopnje poetičnosti in so krajša od navadnih besedil, kjer se po navadi izgubi pomen oz. vsebina teksta. Iz tega lahko sklepamo, da bi lahko vsak poskus izdelave posvečene poezije lahko uporabili kot posvetilo. Vedeti moramo samo ime osebe oz. geslo, ki ga uporabimo, in tematiko oz. namen posvetila. Vstavimo lahko tudi opis osebe in tako še bolj personaliziramo vsebino.

Danny Britz[3][1] trdi, da je umetna inteligenca v zadnjih desetih letih zelo napredovala s prihodom nevronskih mrež. Na začetku jih ljudje nismo zares znali upravljati in čeprav so bile v teoriji odlične, so bili rezultati v praksi večinoma podpovprečni, saj ponavljajoča se nevronska mreža (v nadaljevanju: RNN) ni znala povezati besed, ki so bile več korakov narazen. Tako nismo dobili smiselnega teksta, ali pa je bil poln slovničnih napak.

Glavni problem je bil "vanishing gradient problem", zaradi katerega so razvili LSTM in GRU, katerih arhitektura dopušča dobro povezavo med besedami, ki ne stojijo ena ob drugi v povedi. Vsebujeta namreč črno skrinjico, v kateri so spomin prejšnjih besed in vhodni podatki. Od obeh vzame določen odstotek, ki se ga nauči preko parametrov in učnih primerov in izračuna izhodne podatke.

Ob tem se je tudi pohitril proces izdelave modela oz. učenja RNN in hitrost procesiranja rezultatov, ki se izvaja na grafični enoti. Te iz leta v leto omogočajo hitrejše računanje. Dandanes lahko treniramo RNN od pet do osem ur, točna vrednost je odvisna od problema in velikosti učnih podatkov, in bomo dobili solidne rezultate, medtem ko je ta proces pred 10-15 leti trajal tudi več tednov, da smo dobili spodoben rezultat učenja.

Glavna prednost RNN je, da se bo ne glede na to, kakšne podatke mu podamo, vedno naredil model podatkov in glede na vzorce na vhodu poskušal zgenerirati podobno besedilo. Tako lahko uporabimo tekst v poljubnem jeziku, ga glede na korpus pregledamo in označimo s pomočjo NLTK, ki razvrsti besede glede na besedno vrsto ter začnemo učiti RNN. Andrej Karpathy[15][2] je podal nekaj zanimivih primerov, kjer je generiral zelo raznolik spekter besedil - od algebraičnega dokaza do teksta za dramo Williama Shakespeara.

[1] https://dennybritz.com/posts/wildml/recurrent-neural-networks-tutorial-part-1/
[2] http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Tabela 1: Primer generiranega besedila za Shakespearovo delo (Vir: [15]).

PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

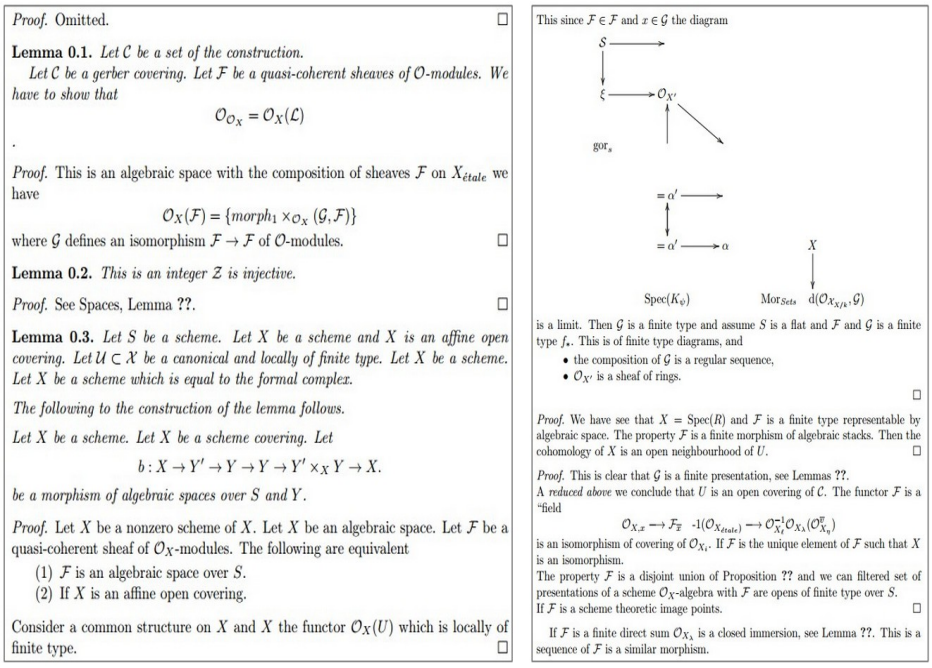
Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.



Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*
Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram

$$\begin{array}{ccc} S & \longrightarrow & \\ \downarrow & & \\ \xi & \longrightarrow & \mathcal{O}_{X'} \\ & \searrow & \uparrow \\ & & \mathrm{gor}_s \end{array} \qquad \begin{array}{ccc} & = \alpha' & \longrightarrow \\ & \updownarrow & \\ = \alpha' & \longrightarrow & \alpha \end{array} \qquad \begin{array}{c} X \\ \downarrow \end{array}$$

$$\mathrm{Spec}(K_\psi) \qquad \mathrm{Mor}_{Sets} \quad \mathrm{d}(\mathcal{O}_{X_{fk}}, \mathcal{G})$$

is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \quad \text{-1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_t}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{x}})$$

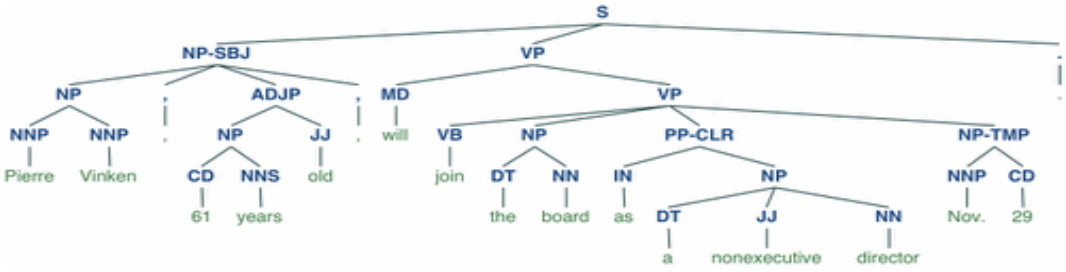
is an isomorphism of covering of $\mathcal{O}_{X_t}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Slika 1: Primer matematičnega dokaza (Vir: [15]).

2 Slovnica

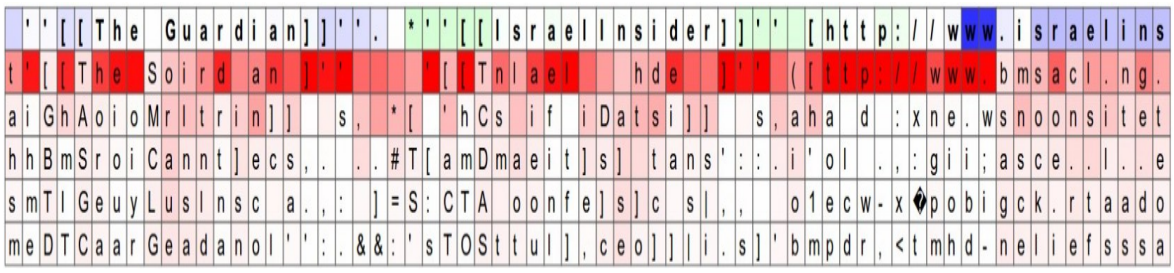
Da bi lahko začeli upravljati in generirati besedila RNN, jih moramo glede na strukturo jezika pravilno razvrstiti v skupine besed. Tako se bo lahko RNN naučila, da za vsakim pridevnikom prihaja samostalnik (oziroma zelo pogosto); ali da vezniki povezujejo stavke, ali so uporabljeni za naštevanje, da so ločila na koncu povedi ipd. Poleg tega vsako besedo tudi razdelimo na morfeme s pomočjo korpusa, saj so rime zgrajene iz istih ali enako zvenečih pripon. S temi koraki označimo in pripravimo vhodne podatke za obdelavo. Na tej stopnji lahko dodatno označimo količino ponovitve besed in se znebimo nekaterih unikatov.



Slika 2: Primer označitve besedila (Vir: [2]).

2.1 Nevronske mreže

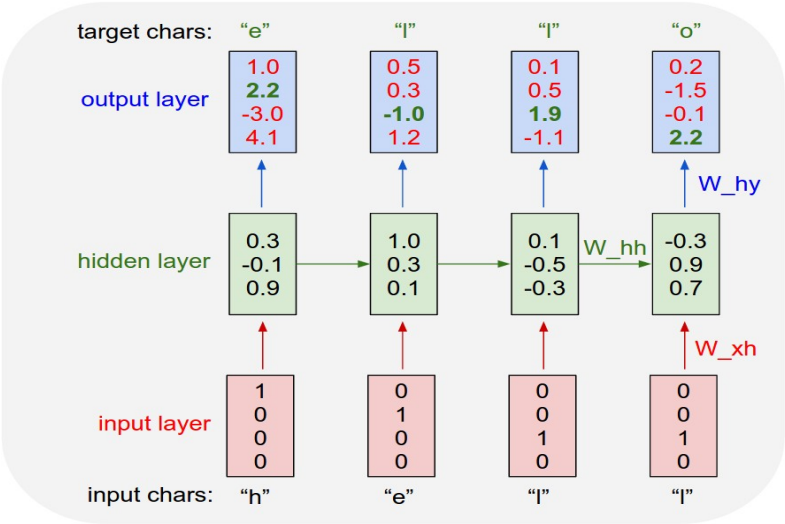
V naslednjem grafu si lahko pogledamo, kako RNN izračuna naslednjo besedo/znak v povedi, kjer je pet kandidatnih črk pri vsakem znaku. Bolj ko je rdeč znak, večje možnosti ima, da se pojavi. Poleg tega imamo v povedi tudi zeleno in modro, ki določata vzorec, ki ga želi najti eden od nevronov v RNN.



Slika 3: Primer iskanja naslednjega znaka (Vir: [15]).

Iz slike lahko razberemo, da želi nevron izvedeti koliko je ponovitev znaka "w", preden se začne naslov URL. Poleg tega lahko vidimo, da je pravilno razbral, da za oglatim oklepajem sledi še en in enako pri apostrofih. Poleg tega vidimo, da se ob drugi ponovitvi obarvata iz modre v zeleno, saj se RNN uči bolje, če je več istih ponovitev nekega vzorca. Po drugi strani se ´IsraelInsider´ spremeni iz nevtralnega v nezaželenega, saj je unikat in se najbrž ni pojavil nikjer drugje v vhodnem besedilu.

Če natančno pogledamo, kaj se dogaja pri vsakem znaku, ki ga program prebere, vidimo, da je vmes še skrito stanje, ki se spreminja glede na vhodne podatke. Vektorji vsebujejo verjetnostno porazdelitev možnosti naslednjega znaka. Pri naslednjem primeru imamo vhodno abecedo sestavljeno iz štirih črk "helo" in želimo naučiti RNN besedo "hello".

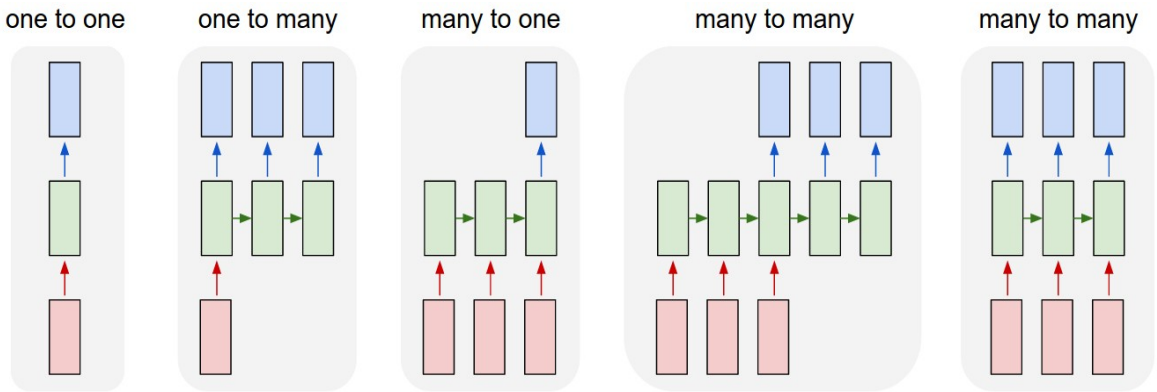


Slika 4: Primer iskanja naslednje črke (Vir: [15]).

Na izhodu želimo imeti zeleno številko najvišjo, saj nam ta pove naslednjo črko z največjo verjetnostjo. Vidimo, da v prvih dveh primerih to ne drži in zato uporabimo BPTT tehniko(Backpropagation Through Time), ki rekurzivno pošilja podatke nazaj, do prejšnjih stanj, kjer se nato sčasoma začne višati zelena številka in ostale padati. BPTT se uporablja dokler ne dobimo konsistentnih številk z učno množico in s tem tudi dobimo pravilno napoved naslednje črke. Poleg tega lahko vidimo da pri prvi črki "l" pričakujemo ponovitev črke, medtem ko v naslednjem koraku pričakujemo "o". Zaradi tega ne moremo upoštevati samo vhodnih podatkov ampak tudi vsebino prejšnjih korakov. Ob tem pa je treba tudi omeniti, da večja kot je učna množica podatkov, tem bolj se lahko približamo želenemu rezultatu, saj s tem RNN dobi več možnih vzorcev vhodnih podatkov, ki jih želimo uveljaviti v končni produkt. Zato učimo RNN večinoma na podatkih iz Wikipedije ali korpusov, kjer je zbranih ogromno besedil različnih zvrsti.

3 Predstavitev raziskovalnega področja

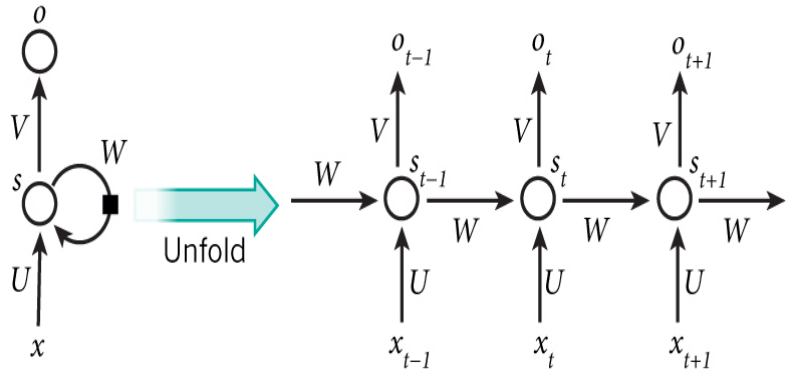
Poznamo več vrst nevronskih mrež. Tiste najbolj osnovne dobijo na vhod podatke, jih preračunajo glede na vmesno stanje in jih vrnejo na izhod. Tedaj govorimo o povezavi "ena proti ena", kjer so podatki neodvisni med seboj. Pri povezavi "ena proti mnogo" vhodne podatke preberemo in jih razčlenimo na več delov, npr. spisati pesem nekomu v čast. Sledi obratna operacija, kjer želimo izvedeti, ali je bilo neko besedilo pozitivno, negativno ali nevtralno. Pri povezavi "mnogo proti mnogo" se začne izhod, ko dobi vse podatke. Ta tip po navadi uporabljamo za prevode, saj so v različnih jezikih drugačno razporejene besede v stavkih. Zato ne moremo prevesti neposredno besede za besedo brez konteksta, saj mora biti stavek pravilno sestavljen. Na koncu ostane še druga različica RNN, kjer so podatki odvisni od predhodnih, obratno pa ne. Tukaj si lahko predstavljamo nek video, kjer je pomembno, da je vrstni red slik pravilen, saj tako dobimo smiseln vzorec.



Slika 5: Primeri osnovnih neronskih mrež (Vir: [15]).

Vsak kvadrat je vektor verjetnostne porazdelitve podatkov in vsaka puščica predstavlja množenje matrik. Vhodni podatki so rdeči, skriti nivoji zelena in izhodni podatki so modri kot pri sliki 4.

Če pogledamo, kako se matrike množijo med seboj, imamo: U, ki označuje vhod; W je spomin prejšnjega nevrona in V je izhod.



Slika 6: Primer računanja v RNN (Vir: [3]).

Za izračun vsakega stanja uporabljamo naslednji formuli:

$$s_t = \tanh(Ux_t + Ws_{t-1})$$

$$o_t = \text{softmax}(Vs_t)$$

Pri izračunu stanja, se pojavi tanh funkcija, ki je najbolj razširjena, lahko pa bi se uporabila tudi katerakoli druga pragovna funkcija. S funkcijo "softmax" pretvorimo surove podatke na izhodu v normalizirane vrednosti od 0 do 1, kjer je skupek vseh točno 1. Adrian Rosebrock[25] v svojem članku natančno razloži, zakaj in kako moramo to storiti. Želimo se ukvarjati z verjetnostmi pojavitve in tukaj se lahko vrednosti nahajajo od 0 do 1, saj ne moremo biti manj kot 0 % ali več kot 100 % prepričani, da se bo znak pojavil.

Pri tehnologiji LSTM in GRU se s_t izračuna drugače, saj imamo poleg vhodnih podatkov še vrata "pozabi", ki povedo, koliko spomina prejšnjega nevrona želimo ohraniti. To pomeni, da lahko izluščimo šumne podatke in dobimo bolj natančne rezultate. Če bi vedno pozabili spomin prejšnje iteracije, bi dobili klasično RNN, kjer smo odvisni le od najbližjih sosednjih znakov/besed.

4 Metodologija

LSTM in GRU sta najbolj napredni nevronski celici, saj imata svoj spomin v vsaki iteraciji. Ta beleži vse spremembe od začetka do konca obdelave podatkov in glede na njih primerno popravi vsebino izhoda. Več kot je tematsko podobnih besed, bolj natančen bo rezultat. Ilya Sutskever in sodelavci[11] trdijo, da nekatere bolj obskurne ali prekrivajoče teme lahko zmedejo RNN in tako dobimo mešane ali nepovezane povedi. Danny Britz[3] trdi, da obe tudi odpravita problem izginjajočega gradienta, ki se pojavi pri preprostih RNN. Tam se zaradi funkcije "tanh" pojavi problem pri določitvi parametrov. Te si nevroni med seboj izmenjujejo, ko pa so spremembe ničelne, se lahko celotno učenje ustavi.

Poleg tega imata algoritem BTPP, ki je drugačen in boljši od navadnega pošiljanja podatkov prejšnjemu procesu. Glavna razlika je, da se gradienti v vsakem procesu seštejejo, preden se pošljejo naprej in tako dobimo manj napak in izgub med nevroni. Christopher Olah[23] trdi, da lahko BPTT pohitri učenje za več ur ali pa celo dni.

4.1 Nevronska mreža na osnovi LSTM

Celica LSTM ima tri vrste vrat - vhod, pozabi in izhod. Poleg tega se naši izhodni podatki (h_t) izračunajo malo drugače, saj moramo v račun všteti omenjena vrata, ki podajo precej več informacij glede možnega izhoda.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t\text{-}1} + W_{ci}c_{t\text{-}1} + b_i)$$

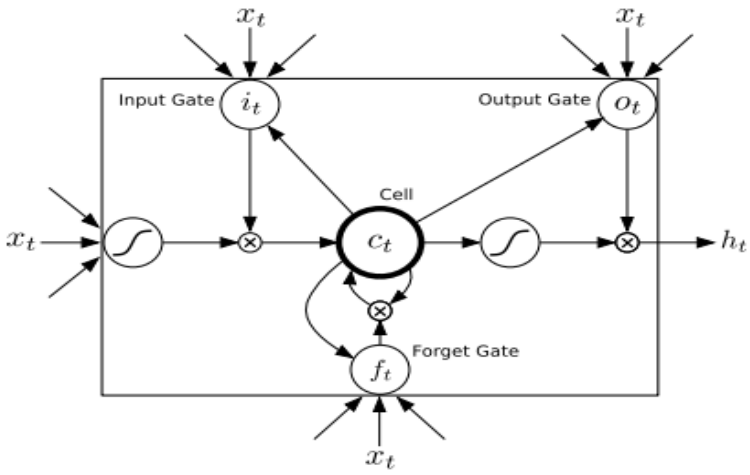
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t\text{-}1} + W_{cf}c_{t\text{-}1} + b_f)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t\text{-}1} + W_{co}c_t + b_o)$$

$$c_t = f_t c_{t\text{-}1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t\text{-}1} + b_c)$$

$$h_t = o_t \tanh(c_t)$$

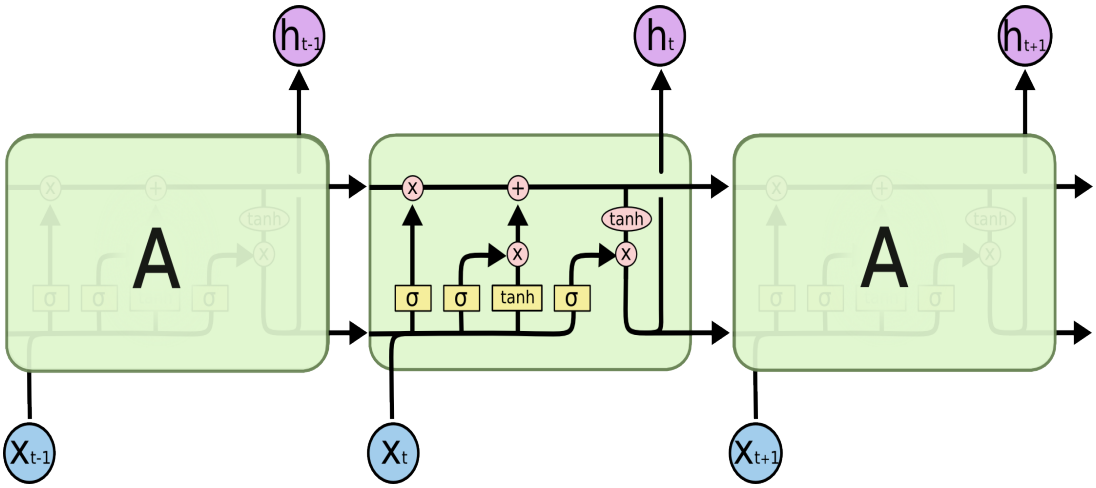
Vidimo, da so na vseh delih enake formule, kjer W prikazuje podatke prejšnje celice in x_t vhodne podatke, ampak vseeno vsak posebej spremeni "svoj" del podatkov. Na vhodnih vratih se odločimo, koliko podatkov bomo sprejeli ali zavrgli. Nato na pozabljenih pretehtamo, koliko bodo novi podatki spremenili spomin, ki ga je prejel



Slika 7: Struktura LSTM (Vir: [7]).

od prejšnje celice, in na koncu na izhodna vrata postavimo posodobljene parametre. Izhodni tekst je kombinacija prvih dveh vrat.

Pri vsaki formuli vidimo tudi σ oz. plast "sigmund", ki oznanja koliko podatkov naj spusti skozi, in deluje kot logična vrata. To je glavni faktor oz. sito, kjer izluščimo pomembne podatke iz pustega besedila.

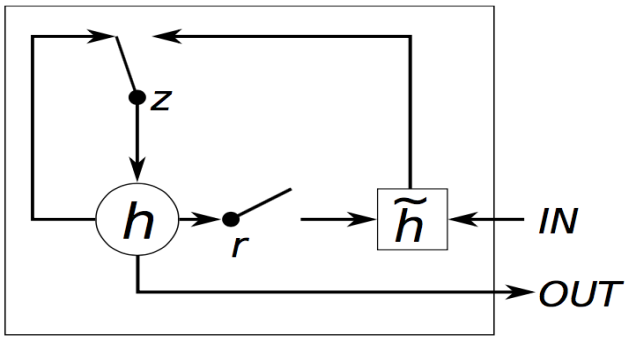


Slika 8: Veriga LSTM (Vir: [22]).

Veriga je dolžine t in ko dobimo vrednosti vsake celice c_t, se proces ustavi. Vsak izhod bo vrnil poved in tako dobimo t-vrstic besedila. Pri učenju se konec verige sklene z začetkom in tako se cikel ponovi neštetokrat. V vsaki iteraciji dobimo bolj natančne rezultate, ampak če bi pustili RNN, da se uči predolgo, bi se slabo odzval na nove podatke, saj bi bil preveč prilagojen na že obstoječe.

4.2 Nevronska mreža na osnovi GRU

Celica GRU ima samo dvoje vrat - ponastavitev in posodobitev. Prva vrata pokrijejo vhodna in vrata "pozabi" v modelu LSTM. Poleg tega sta združena tudi c_t in h_t, ki ponazarjata vrednost celice in skrito stanje. Rahul Dey in Fathi M. Salem[6] trdita, da v večini primerov, kjer imamo kompleksne teme in se ne oziramo toliko na sentimentalnost teksta, GRU dosega boljše rezultate, saj je bolj specifičen.



Slika 9: Struktura GRU (Vir: [12]).

$$r_t = \sigma(W_r x_t + U_r \hat{h}_{t\text{-}1})$$
$$z_t = \sigma(W_z x_t + U_z \hat{h}_{t\text{-}1})$$
$$\overline{h}_t = \tanh(W x_t + U(r_t \odot \hat{h}_{t\text{-}1})$$
$$\hat{h}_t = (1 - z_t) \odot \hat{h}_{t\text{-}1} + z_t \odot \overline{h}_t$$

Če je GRU boljši od LSTM v specifičnosti, ga ta prekaša v občutljivosti podatkov. Na splošno pri večji količini podatkov GRU boljše prepozna odvečne podatke in jih odstrani. To je njegova največja prednost, saj želimo imeti čim večjo učno množico. Poleg tega se zaradi bolj preproste strukture hitreje uči. [18, 21]

4.3 Enota za procesiranje naravnega jezika (NLP)

Da se lahko računalnik sporazume z osebo in tako zaživi umetna inteligenca, je ključnega pomena, da nas le-ta razume. Problem pri uporabi avdio vsebin je, da se govorni jezik precej razlikuje od knjižnega. Ob tem imajo avdio posnetki vedno ogromno šuma, razen če so posneti v profesionalnih studiih in s profesionalno opremo. Poleg tega imamo še govorna narečja, kjer spet vsak drugače naglasi različne besede in je od vasi do vasi drugačen jezik. Zato je NLP večinoma uporabljena za procesiranje že napisanih besedil.

Zbirke besedila najdemo v korpusih, kjer je shranjenih veliko večino časopisov, revij, spletnih besedil, leposlovja, literature, šolskih učbenikov in podobnih besedil. Za primer lahko vzamemo največji slovenski korpus Gigafida[20][1], ki ima zbranih 38.310 besedil, v katerih je vse skupaj več kot milijardo besed (1.134.693.933).

NLP se najbolj osredotoča na sintaktične in semantične analize teksta. Z njo upravljamo z naslednjimi razpoložljivimi programskimi orodji:

- NLTK,
- Stanford toolkit,
- Gensim,
- Open NLP.

4.3.1 Sintaktična analiza

Great Learning Team[26] trdi, da sintaktična analiza vsebuje sedem glavnih tehnik.

- Prva je lematizacija, kjer več izpeljank združimo v isto lemo (besedni razred). Tukaj poleg vseh besed z istim korenom dodamo še stopnjevanja (npr. boljši spada pod lemo dober).
- Sledi morfološka segmentacija, kjer razdelimo besede na posamezne morfeme. Uspešnost procesa je odvisna od izbire jezika, saj imamo npr. angleščino, kjer s pregibanjem besed razčlenimo skoraj vse besede. V kolikor je možnih oblik besede malo, jih lahko uporabimo kot ločene besede. Na drugi strani pa imamo Indijski jezik, kjer ima lahko beseda več kot 1000 možnih besednih oblik in je postopek manj uspešen.
- Nato razdelimo besedilo na besede. Tukaj glede na presledek dobimo vse besede v slovenščini. Pri angleščini moramo upoštevati še apostrof, saj le-ta združuje dve besedi.
- Za tem imamo govorno označevanje, kjer v besedilu označimo besedne vrste. Tako točno vemo kakšen pomen ima neka beseda in kako je uporabljena v povedi.
- Sledi razčlenjevanje odvisnosti in gramatike. V tem koraku se zgradi drevo razčlenitve besednih vrst, ki ga Dan Klein in Christopher D. Manning[16] v njuni raziskavi bolj podrobno opišeta.

[1] https://viri.cjvt.si/gigafida/

- Pri razdeljevanju besedila na stavke moramo paziti, kdaj je končno ločilo uporabljeno za konec povedi, kdaj pa se lahko znajde tudi sredi povedi (pika se uporablja za okrajšavo besed: "npr.").

- Na koncu ostane iskanje korena oz. krnjenje. M. Juršič[13] trdi, da pri slovenščini krnjenje doseže precej slabe rezultate, saj pogosto dobimo prekratke korene in se zaradi tega preveč besed preplete in izgubimo informacije v besedilu.

4.3.2 Semantična analiza

Pri semantični analizi se večinoma uporabljajo tri glavne tehnike. Prva je iskanje imen in razvrstitev le-teh v skupine. Te so npr. ime osebe, kraja, podjetja, dogodka... Ta proces je pomemben, saj so v različnih jezikih lahko nekatera imena napisana z malo začetnico in bi tako lahko prišlo do napake pri označevanju. Največkrat se to lahko zgodi ob imenih, ki že imajo same po sebi pomen, npr. podjetja Apple ne želimo obravnavati kot jabolko. Naslednja tehnika je določitev pravega pomena besed, npr. "pes je zbežal" in "pes je meso", kjer ima beseda "je" dva pomena. Pomembno je tudi razbrati, ali gre za prenesen pomen ali pa dobeseden (npr. dobil je kurjo polt). Ostane še tehnika izdelave rezultata, ki s pomočjo ogromne baze podatkov semantičnih lastnosti pretvori podatke v smiseln zapis, ki ga lahko ljudje razberemo. [26]

4.4 Modul za izdelavo besedila

Pri izdelavi modula za besedilo moramo biti pri izbiri učnih podatkov zelo pazljivi. Problem se lahko pojavi, ko izberemo preveč besedil, ki imajo ogromno unikatnih besed, ki se v vsakodnevnem govoru ne pojavijo (npr. Wikipedija). J. Hopkins in D. Keila [10] sta preizkusila uspešnost pretvorbe besedila v foneme in nazaj in tako ugotovila, katera učna množica je boljša za uporabo.

Tabela 2: Primerjava uspešnosti pretvorbe besedila na foneme in nazaj (Vir: [10]).

	Besede	Povedi	Pokritost
Wikipedija	64.84%	83.35%	97.53%
Soneti	85.95%	80.32%	99.36%

Kakor vidimo v zgornji tabeli, so soneti precej bolj uspešni. Pri pretvorbi besede za besedo je več preprostih besed glede na Wikipedijo, kjer je ogromno unikatnih in prekrivajočih se besed. Medtem je pri pretvorbi povedi malce boljša Wikipedija, saj se tukaj gleda "trigram" podobnost. Tukaj se primerjajo vsake tri besede, ki si sledijo.

Soneti v tej primerjavi izgubijo, saj so tam besedila težje berljiva od navadnega besedila. Zadnja primerjava prikazuje delež pokritosti besed v slovarju.

Pri kreiranju modulov moramo biti striktni, saj lahko umetna inteligenca generira pesem, ne da bi ji podali katerikoli parameter. Običajno podamo temo, saj tako dobi pesem vsaj nek smisel oz. idejo. Daniel W. Otter in sodelavci[4] trdijo, da lahko poleg pesmi pod to kategorijo uvrstimo tudi šale in zgodbe. Vseeno je pesem najtežja, saj smo omejeni s strukturo, ritmom in poetičnostjo.

Pri šalah je največji problem, da so večinoma kratke in da so bili prvi testi dokaj slabi, saj so se posvetili samo večpomenskim in podobno zvenečim besedam. Nato so z učenjem na zbirki šal in novicah dodali dodaten kontekst in signifikantno izboljšali rezultat.

Pri zgodbah je največji problem pri ohranjanju povezanosti in ustvarjalnosti. Umetna inteligenca nikoli ne ve, kdaj zaključiti zgodbo in tako vedno ohranja več možnih scenarijev. Poleg tega so zgodbe, ki se pripovedujejo, bolj uspešne od dialoga ali scenarija, kjer je tekst razdeljen glede na vloge.

4.5 Modul za izdelavo rim

Preden nam RNN izpiše rezultat, lahko z vključitvijo rim in tem izboljšamo želen rezultat, saj izberemo vrstice, ki so si po vsebini podobne. To je ključnega pomena, saj bi v nasprotnem primeru dobili generator naključnih vrstic besedila.

Rime se po navadi pojavljajo na koncu vrstic, zato pri njihovi implementaciji uporabljamo izdelavo besedila v obratnem vrstnem redu. Tako lahko vedno "nastavimo" rimo in odstranimo vse stavke, ki so drugače oblikovani. Če bi poskusili postaviti rime na konec vsake vrstice, bi dobili precej nelogičnih stavkov. R. A. Kai-Ling Lo in P. Kurz [14] sta naučila RNN in to trditev potrdila, saj jima ni uspelo ustvariti rime AABBA v prvotnem zaporedju.

V kolikor bi želeli, da bi se rima pojavila še kje drugje v besedilu, Lanqing Xue in sodelavci[19] trdijo, da bi morali vključiti tabelo vseh fonemov oz. kandidatov rim, ki se pojavijo na koncu vsake besede. S pomočjo tabele bi potem glede na isti položaj v vsaki želeni vrstici postavili ujemajočo besedo. V kolikor zanemarimo položaj v tekstu, se ritem in berljivost precej poslabšata. Če bi želeli, dobiti besedilo za rap pesem in bi želeli da se rima več kot samo zadnji fonem, pa moramo vedeti, ali želimo, da se besede ponavljajo ali ne. Tukaj lahko namreč dostikrat zmanjka razpoložljivih besed, ki bi se ujemale.

4.6 Teme

Pri izbiri teme dobimo najboljše rezultate, če vključimo nekaj najbližjih tem skupaj, saj tako razširimo skupek besed, ki jih RNN uporabi. Tako večinoma ohranimo vsebinsko enake vrstice. Poleg tega želimo izbrati čim bolj opisljive in širše teme, saj se tako ne omejimo preveč.

J. K. Brendan Bena [1] je ustvaril pesmi glede na emocije in prišel do zaključka, da je lažje nevronski mreži razbrati in napisati pesem o osnovnih emocijah (sreča, žalost, jeza) kot o bolj kompleksnih, ki jih je težje opisati.

Tabela 3: Uspešnost razbiranja emocij v pesmih (Vir: [1]).

Emocije	Jeza	Pričakovanje	Sreča	Žalost	Zaupanje
%	65	40	85	87.5	32.5

Da bi lahko rime in teme vplivale na razplet generacije stavka, morata biti obe normalizirani na skalo od 0 do 1, saj so podatki v RNN porazdeljeni na verjetnost pojavitve.

Glede na to, da želimo oblikovati pesem podobno temam, na katerih se RNN uči, je ključnega pomena, da učno množico tematsko pravilno označimo, saj bi drugače prišlo do šuma pri generaciji. Ta se pojavi, ker so učni podatki slabo označeni. Zato moramo uporabiti še en model RNN za sortiranje vhodnih pesmi. Rajat Agarwal in Katharina Kann [24] sta namreč uspešno učila 60.000 pesmi brez teme z manjšo množico tematsko sortiranih pesmi. Pri evaluaciji sta spoznala, da lahko celo množica brez vnaprej definirane teme doseže primerljive rezultate.

5 Implementacija

Osnovna podlaga programa je last Tima Van de Cruysa [5], ki je ustvaril generator pesmi iz vsakdanjih besedil. Odločili smo se, da bomo uporabili njegovo podlago, saj nas je zanimalo, do katere mere lahko navadno pusto besedilo razvlečemo, da bo še vedno zvenelo kot pesem. Presenetila me je uspešnost programa, saj so vhodni podatki brez nekih globljih idej in originalnosti, medtem ko so se najboljši rezultati precej približali človeški poeziji.

Obenem je poleg angleščine učil RNN tudi za francoščino. To je bilo ključnega pomena, saj tako lahko dobimo primerjavo med dvema jezikoma, ki sta učena na isti nevronski mreži z istimi parametri. Tako lahko ugotovimo, kateri jezik je boljši za generacijo in bi lahko v nadaljnji raziskavi vključili še več jezikov in našli najboljšega.

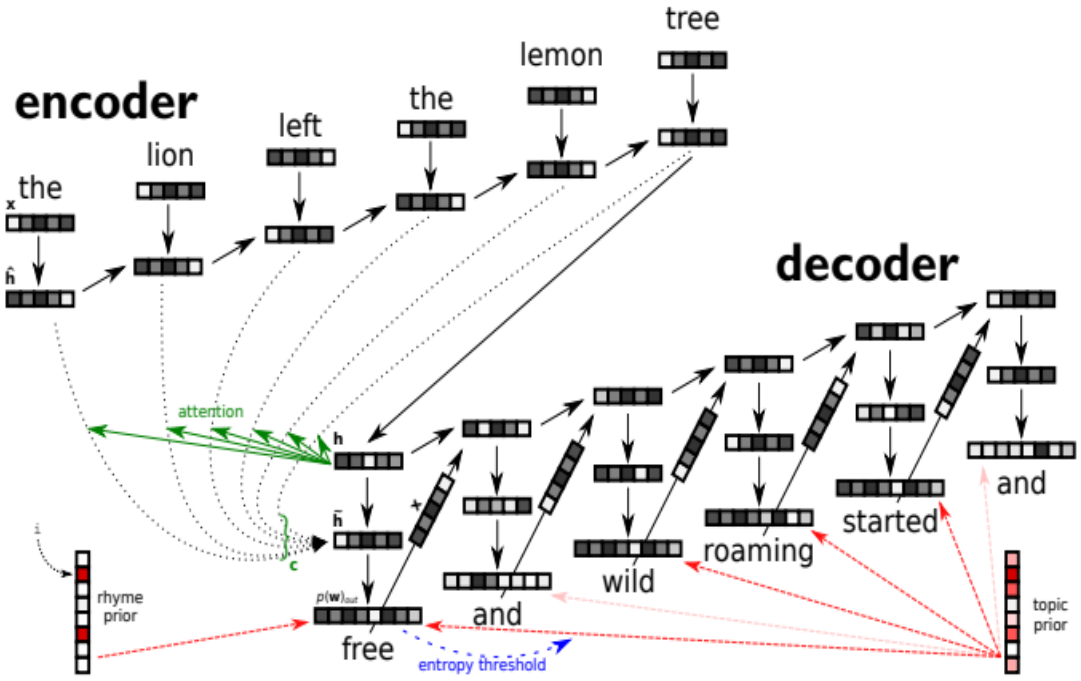
Primerjali smo rezultate generiranih pesmi in posvetil. V anketo smo vključili najboljše primerke, ki jih je izbrana množica, ki tekoče govori jezik, ocenila glede na pomen, poetičnost in gramatično pravilnost. Poleg tega so anketiranci ocenili, kako naravno ali pa robotsko se posvetilo/pesem sliši.

5.1 Opis programa

Program je napisan v programskem jeziku Python, ki ga je ustvaril Guido van Rossum leta 1990. Pri implementaciji nevronske arhitekture, sta uporabljeni programski knjižnici PyTorch in OpenNMT. Guillaume Klein in sodelavci[17] trdijo, da je slednja ključnega pomena, saj se ukvarja s prevajanjem besedila in tako služi kot most med različnimi jeziki in nevronsko mrežo. Poleg tega lahko omenimo še knjižnico KenLM, ki pohitri računanje izdelave in dekodiranja modela RNN in ob tem tudi uporabi manj spomina. K. Heafield[8, 9], ki je razvil knjižnico, trdi, da je v primerjavi s splošno znanim SRILM 2.4-krat hitrejši in uporabi 43 % manj spomina.

Model RNN je izdelan s pomočjo kodirnika in dekodirnika. Kodirnik zgradi prvo vrstico, jo kriptira in pošlje dekodirniku. Ta naslednjo vrstico zgradi v obratnem vrstnem redu, saj je glavna prioriteta rima, ki jo najprej naredi (v kolikor se le-ta takoj ponovi). Ob tem na vsako besedo vpliva tudi tematski vektor, ki skupaj drži vsebino vrstice v naprej določeni tematiki. Slednji močneje vpliva na samostalnike, pridevnike in glagole, ki jih bolj uskladi, medtem ko so vezniki in ostale besedne zveze tematsko

nevtralne, saj jih najdemo v vsakem besedilu.



Slika 10: Grafična struktura generatorja pesmi (Vir: [5]).

5.1.1 Vsebina modela GRU

Kodirnik in dekodirnik sta sestavljena iz dveh modelov GRU. Vsak ima skrito stanje velikosti 2048 in vektorje besed velikosti 512. Slednji so deljeni med kodirnikom, dekodirnikom in izhodom. Parametri so optimizirani s pomočjo stohastičnega gradientnega spusta s prvotnim učenjem 0,2, ki se potem zmanjša na četrtino, ko izguba ne izboljša več validacijskega seta.

Za učenje se uporablja velikost serije 64, nad katero je uporabljena metoda gradientnega obrezovanja, ki spreminja določitev napake, ali pa spremeni prag napake pri BPTT. Nevronska arhitektura je bila implementirana s pomočjo knjižnice PyTorch in z dodatno podporo modula OpenNMT .

5.1.2 Izbira korpusa za analizo in izbira besedila

Za učenje podatkov se je uporabil korpus CommonCrawl. Pri vsakem modelu je bilo treba odstraniti vse ostale jezike in ohraniti povedi, dolge največ 20 besed, saj se poezija in dolge povedi ne skladajo ena z drugo. Poleg tega filtriramo vse povedi, kjer

ni funkcijskih besed, saj s tem odstranimo vse gramatično nepravilne in problematične povedi. Na koncu se odstrani vse povedi, ki se niso ponovile v dokumentu/korpusu.

Po filtriranju se je naredil učni korpus, ki je vseboval 500 milijonov besed za vsak jezik. Ta se je nato zreduciral na 15 tisoč najbolj pogostih besed. Ostalim besedam so se spremenile vrednosti v znak '<unk>' in možnost pojavitve na 0.

Poleg tega je bil implementiran klasičen trigram model Kneser-Ney , s pomočjo KenLM in pa model NMF, ki ima 100 dimenzij. V vsaki od njih so vsebovane tri besede oz. teme, ki jih potem RNN uporabi pri generiranju posvetila. Oba modela sta bila naučena na ogromnem korpusu, ki vsebuje 10 bilijard besed. Slednji je bil filtriran samo glede na jezik, ki smo ga potrebovali. Za Gaussovo porazdelitev je bilo izbrano povprečje 12 in standardna deviacija 2. Parameter vpliva topike na izhod je bil nastavljen na 2,7.

5.2 Priprava vhodnih podatkov

Za vsako vrstico je generiranih približno 2000 različnih kandidatov, ki vsi sledijo rimski shemi in podani tematiki. Sortirani so glede na možnost pojavitve in povezanosti s prejšnjimi vrsticami. Zaradi tega lahko akrostih včasih zelo pokvari kakovost posvetila, saj lahko namesto najprimernejše vrstice dobimo petstoti najboljši rezultat.

Rime, ki so uporabljene, so razdeljene glede na dolžino posvetila in so naslednje:

- 4: ABAB,
- 5: ABCBA,
- 6: ABAB CC,
- 7: ABAB CDC,
- 8: ABAB CDCD

Programu lahko podamo akrostih in številko teme 0-99, ki so pred vsako generacijo izpisane. V kolikor ne podamo prvega, bo program za akrostih uporabil besedo "poet". Pri temi naključno izbere temo s seznama.

5.2.1 Primeri končnega produkta

Tabela 4: Primeri posvetil.

Angleščina	Francoščina
Gradually the world ranks among those we find and today the world has become great perhaps many people own our own mind in fact i considered living my own fate	Non , il ne faut pas que ça arrive ils sont tous là , c' est tout ce qui compte ce sont tous des gens qui nous suivent hélas , il ne faut pas qu' ils montent
Pale lips, lips look crooked, from shiny hair to dark skin oh, i dont smoke, the boy replied emily took a deep breath and let the chill sink in then she closed the door and climbed inside	Parlons de la marque des fauteuils optez pour une décoration simple et brève envie de décorer vos meubles en un clin d' œil mais c' est aussi l' occasion de réaliser votre rêve
push it out of the way and see if you wish open the door and see if it will come in excuse me, she said in spanish this is a funny thing, she said with a grin	Pour lui , il fallait faire équipe oublier de le laisser s' emparer du trône et cela est loin d' être le seul principe mais il est possible de changer de zones

Pri francoskih rezultatih se pojavi problem pri črki 'k', saj obstaja v slovarju le nekaj 10 besed, ki se s to črko začnejo. Tako so vsa imena, ki vsebujejo 'k', avtomatsko v podrejenem položaju in bi lahko glede na temo že skoraj določili besedo, ki se bo uporabila. Ob tem pa so tudi rime slabše, v kolikor je zadnja beseda daljša, saj se gleda le zadnji fonem v vsaki vrstici, ne pa celotne besede.

5.3 Raziskava

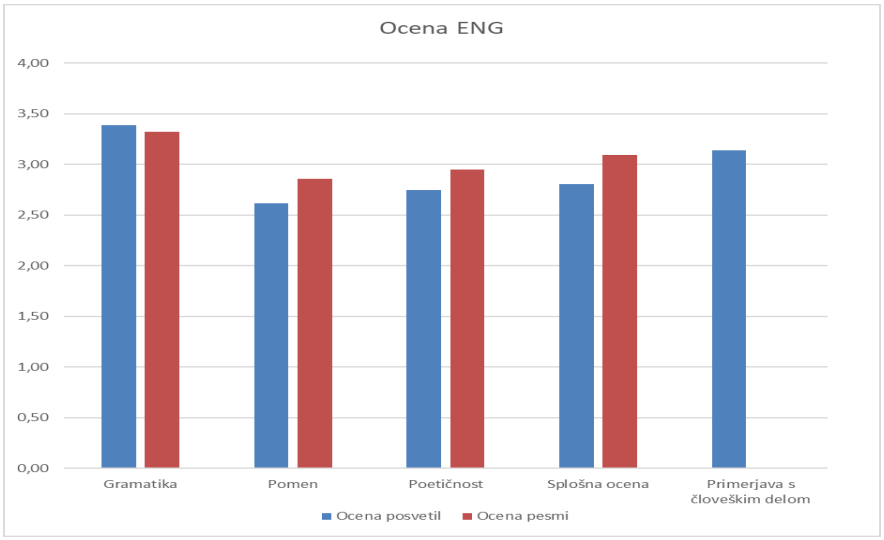
Da bi ugotovili kakovost posvetil, je bilo treba izvesti anketo za francoščino in angleščino. V njej je bilo vsebovanih najboljših šest posvetil, ki so bila ročno izbrana iz testiranja. Ob tem so bile dodane še štiri pesmi brez akrostiha. Tukaj je bila kakovost precej boljša, zato ni bilo treba uporabiti več primerov.

Vsako posvetilo je bilo treba oceniti 1-5 glede na gramatiko, pomen in poetičnost ter podati splošno oceno. Nato je bilo treba še dodatno oceniti, kako bi posvetilo uvrstili na lestvico od robotskega teksta do človeške poezije. Ob koncu vprašalnika so anketiranci podali še njihovo znanje angleškega oz. francoskega jezika ter podali svoja mnenja in opazke. Obe anketi je rešilo 15 tekoče govorečih ljudi, pri angleščini pa smo vključili še dodatnih 10 s srednješolskim znanjem jezika.

Ob tem smo se odločili tudi vključiti rezultate raziskave generiranja pesmi, ki jo je naredil Tim Van de Cruys[5] v začetnem projektu.

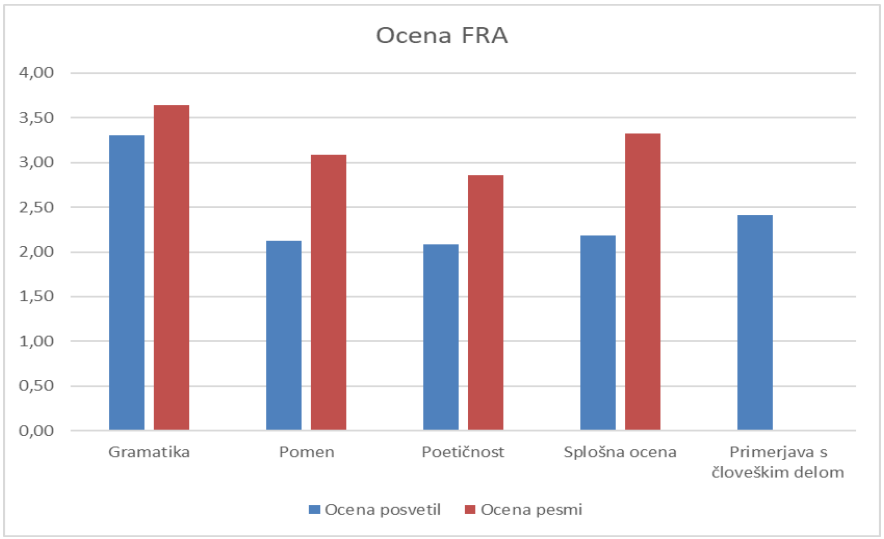
5.4 Rezultati

Pri gramatiki so se končni rezultati skoraj ujemali s prvotnim generatorjem pesmi oz. so se celo izboljšali iz 3,32 na 3,39, kar pomeni, da ne zmoti končnega rezultata če vnaprej določimo nekatere besede, medtem ko so vsi ostali rezultati pričakovano slabši. Pomen pesmi se iz 2,86 zmanjša za 0,25, ko vključimo akrostih, poetičnost iz 2,95 na 2,75 in splošna ocena iz 3,08 na 2,81.



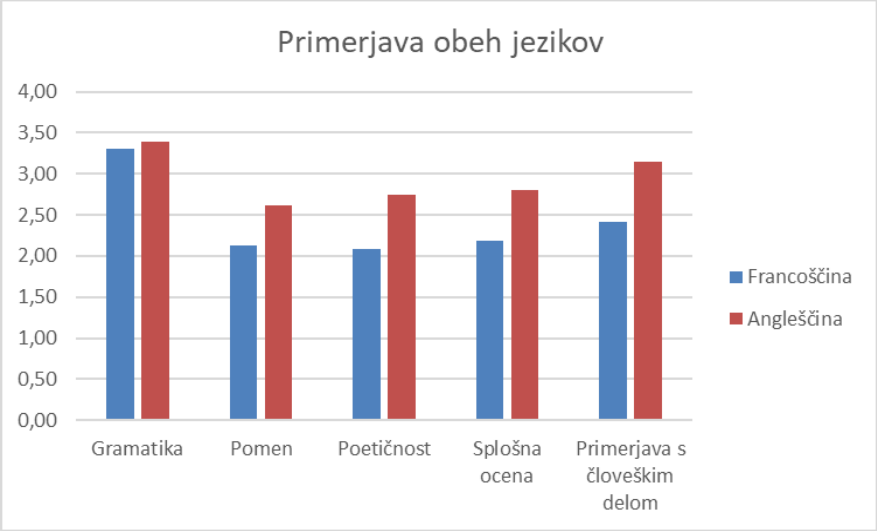
Slika 11: Ocena angleških posvetil.

Pri francoščini lahko opazimo drastični upad vseh vrednosti razen gramatike, čeprav je bilo v prvotnem generatorju pesmi večino rezultatov boljših od angleških. Pomen in splošna ocena tukaj izgubita približno za eno oceno kakovosti (0,94 in 1,13).



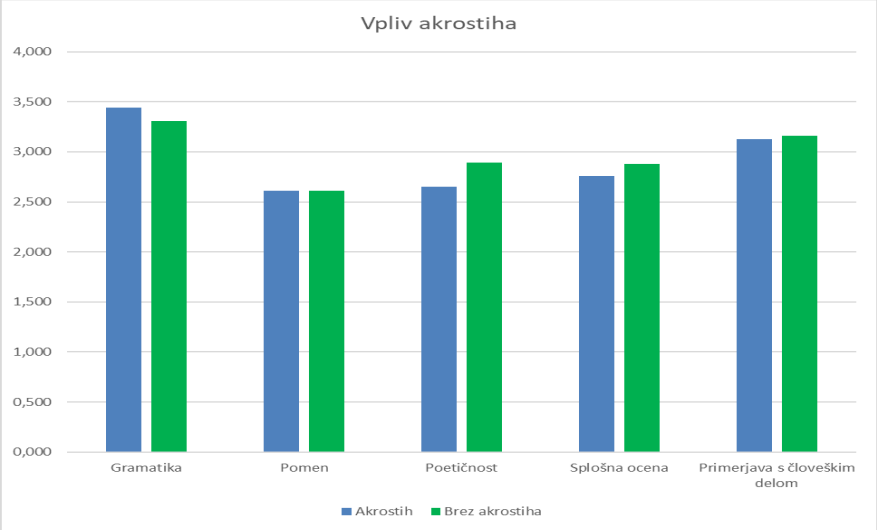
Slika 12: Ocena francoskih posvetil.

Če primerjamo obe verziji, vidimo da sta si gramatično gledano podobni, medtem ko se je pri ostalih francoska različica poslabšala - največ pri primerjavi s človeškim delom, kjer je razlika 0,73, in pri poetičnosti, kjer je zaostala za 0,67.



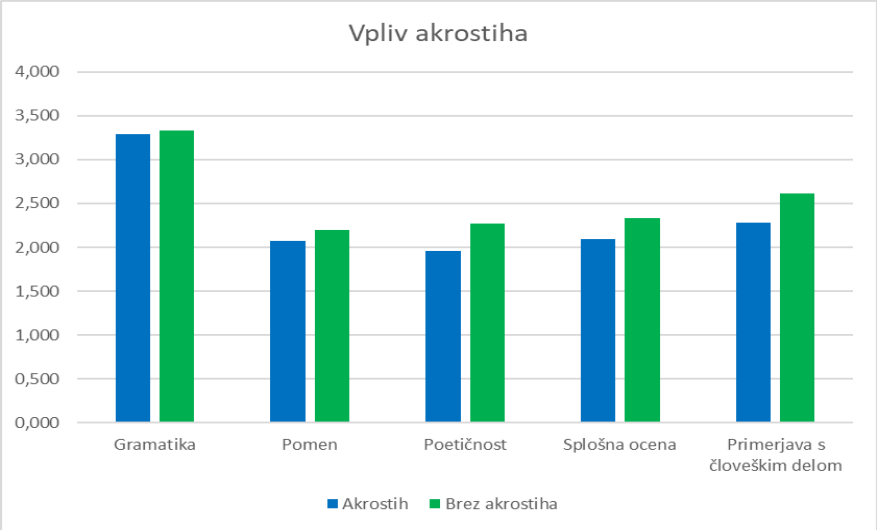
Slika 13: Primerjava obeh verzij.

Nato lahko pogledamo, koliko škode je naredil akrostih glede na ostala posvetila. Vidimo, da so spremembe v kakovosti minimalne, razen pri poetičnosti, kjer se iz 2,89 zmanjša na 2,65.



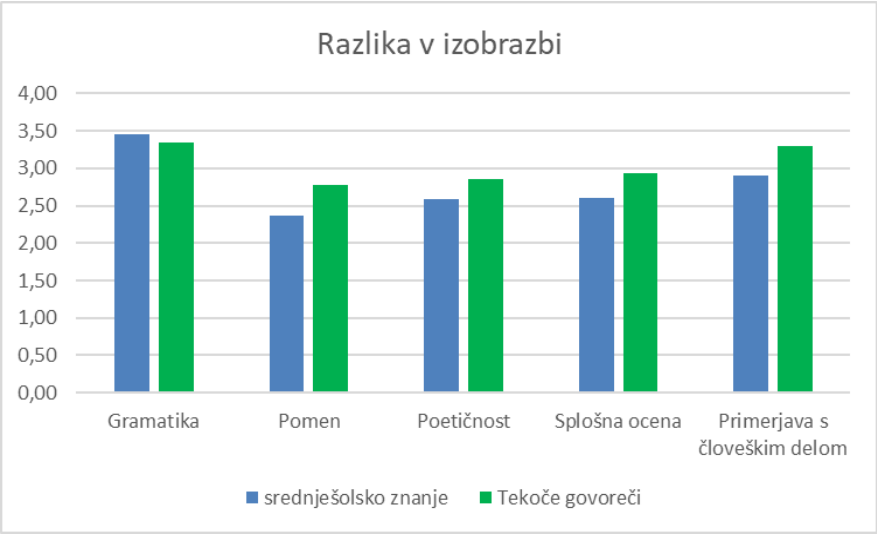
Slika 14: Vpliv akrostiha na angleško posvetilo.

Pri francoščini so razlike malo večje, od 0,26 pri poetičnosti do 0,34 pri končni primerjavi s človeško literaturo.



Slika 15: Vpliv akrostiha na francosko posvetilo.

Razlike glede na izobrazbo oz. razumevanje jezika so tudi razvidne, saj tekoče govoreči bolj razumejo težja besedila in jih zato bolj korektno in boljše ocenijo, ob tem pa najdejo kakšno gramatično napako več.



Slika 16: Razlika glede na izobrazbo.

5.5 Sklep

Iz rezultatov lahko razberemo, da se je kakovost pričakovano poslabšala. Treba je seveda omeniti, da je bila tema v skoraj vseh naključna. Poleg tega nobena izmed njih ni vsebovala nekih čustev, sanj ali misli, ki bi jih lahko abstraktno razumeli. Če bi vključili samo takšne teme, bi ohranili višjo oceno pomena, saj bi imelo besedilo globlji pomen, ki bi ga bilo treba razumeti, ampak se je bolje posvetiti lažjim temam in tako olajšati delo razčlenjevanja posvetila. V nasprotnem primeru bi za vsakega potrebovali vsaj nekaj minut, da si ga nekajkrat preberemo in poskusimo razumeti misel oz. idejo, ki je bila v pesmi izražena.

Izmed posvetil ne bi vsaj eden razvrstil med človeško literaturo tri primere, medtem ko se ta scenarij pri pesmih pojavi le enkrat. Res je tudi, da je večina ostalih primerov dobila samo eno ali dve najvišji oceni, le dve pesmi pa sta prepričali tri anketirance. Pri francoščini so vsi razbrali, da posvetil ni napisal človek, so pa v dveh primerih v večini ocenili, da je tekst dober približek poznani poeziji.

Ob tem je treba omeniti, da je bila izbira francoskih besedil malce težja, saj ne govorijo vsi jezika vsakodnevno in bi se lahko ob generaciji dodatnih 50 posvetil zagotovo našel nek primer, ki bi ga lahko postavili ob bok pesmim. Ne glede na to se je med angleškimi posvetili odločalo, katere naj se ne vključi v anketo, saj je bilo kar nekaj zanimivih. Medtem se je pri francoskih posvetilih našlo samo šest solidnih primerov, ki bi jih lahko napisal človek.

Če bi vzeli najboljše rezultate in jih ročno pregledali ter popravili oz. zamenjali nekaj besed, bi dobili odlične rezultate. Tako bi lahko tudi bolj personalizirali posvetilo, saj so le-ta po navadi nevtralna.

5.6 Možne izboljšave in predlogi

V nadaljnjem delu bi se dalo izboljšati predvsem kakovost, saj je trenutno RNN trenirana na točno tistih podatkih, ki bi se jih morala izogibati. S tem bi nevronska mreža dobila precej boljšo sliko generiranja in bi tako postalo posvetilo bolj poetično, saj je sedaj samo izpisovala naključne prozaične povedi. S tem, da smo ustvarili poizkus na teh podatkih, smo ugotovili, kako dobro poetično posvetilo lahko RNN naredi iz vsakdanjih besedil. V nadaljnji raziskavi bi lahko poskusili ponavljati izdelavo posvetila, dokler vsaka vrstica izpisa ni glede na parametre med najboljšimi dvajsetimi. Tukaj bi lahko trajal proces nekaj minut, po katerih bi bilo treba nastaviti omejitev, v kolikor ne bi RNN uspela najti dovolj dobrega posvetila.

Poleg tega bi se dalo tudi slovnično izboljšati generator, vsaj glede dveh besed, ki jih lahko pišemo skupaj (npr. ca n't), da bi se odstranil presledek. Medtem ko je glede velikih začetnic predvsem pri črki "I" problem, da je program en in deluje za več različnih jezikov. Tako se ne moremo osredotočiti samo na angleščino, saj bi potem vsako francosko besedo, ki bi jo zaznal po angleški slovnici, spremenil in tako naredil napako.

Poleg tega bi odstranjevanje nepotrebnih in slabih tematik precej izboljšalo kakovost. S tem odstranimo tudi nek odstotek možnosti, da program javi napako, da ne more sestaviti naslednje vrstice besedila. Tako bi imela RNN vedno na razpolago emocionalne in odprte teme, kjer je lahko več 10 različnih povedi, ki bi jih lahko uporabili in se ne bi posvetilo pokvarilo.

Seveda bi bilo v nadaljnjem delu mogoče vključiti še več jezikov in tako ugotoviti, če je kateri drug uspešnejši od angleščine. Ko bi se našel nek boljši, bi se lahko iz primerov izvleklo jedro, se ga prevedlo, gramatično in poetično popravilo ter uporabilo, v kolikor ne bi bila kakovost posvetila slabša od primerkov v končnem jeziku.

Tabela 5: Predlogi angleških posvetil

Angleščina
Menim da orodje odlično prevzame nianse izgovorjave besed pri iskanju rim, vendar pa peša pri pravilnem nastavljanju velikosti črk, še posebej "I" in postavljanju presledkov pred "n't".
Generator bi moral vključiti vejice, pike, velike začetnice in apostrofe. Vrstice se med seboj velikokrat ne ujemajo in beseda ne teče med njimi kot v pesmih.
Slovnično je pravilno, problem je, da vse ostalo večino časa nima smisla.
Zdi se, kot da bi se nekdo sproti izmišljeval naključne povedi.
Večino besedil ni bilo slovnično pravilno napisanih.
I naj bo z veliko.
Verzi se mi zdijo nepovezani, napake npr. do n't in ca n't (neki tazga), i z malo začetnico, kakšni deli se mi ne zdijo smiselni.

Francoščina
Večina jih zveni kot reklamni teksti, nekateri pa so popolnoma naključni. Nobenega od njih ne bi štela za klasično literaturo, nekatere pa bi mogoče uporabili v televizijskem oglasu (predvsem prvo). Vse se rimajo (videla sem samo eno slabo "osirotelo" vrstico), vendar vse ne upoštevajo števila zlogov v vsakem verzu.

Kaj je z vejicami, zakaj je pred njimi presledek? Zaradi premalo izkušenj si ne upam soditi gramatike, ampak izpade čudno to, da v isti vrstici piše enkrat cœur, drugič pa coeur. Kar se tiče rimanja, pri 4-vrstičnih kar dobro zadane, sploh glede na to, da se konci besed pišejo na tako različne načine, in ujame izgovorjavo (arrive in suivent se rimata pri iv). Kadar hoče pesniti o kompleksnih temah, ne sestavi pomenov smiselno, vmes rahlo zamenja temo in nastane nekaj brez repa in glave. Še ko vključiš domišljijo, da gre za pesniško jamranje iz srednjega veka, kraljevine in ljudi, povedano nima smisla. Če pa poje o abstraktno enostavnih temah, kjer se tema ne rabi logično stikat, kot je ljubezen, pa smisel pogojno pride skozi. Tam ni veliko za zgrešit, je pa hitro osladno in ponavljajoče. Nekako tako kot človeške pesmi, tako da se je vsaj tukaj približal.
Ko gre za pesmi, so bile nekatere res čudne, saj nisem mogel dojeti pomena ali teme, ki je bila obravnavana. Bili so lepi poskusi, ki pa na koncu niso uspeli, saj proti koncu postanejo precej nenaravni. Po mojem mnenju je AI poskušal uskladiti besede, da bi se rimale, vendar je bila to njegova največja napaka. Kot beseda, ki se je v pesmi zdela najbolj neumestna, je bila v večini primerov rima.
Nekateri so dobri primeri, vendar imajo pogosto kakovost »neželena pošta, ustvarjena z umetno inteligenco«. S tem mislim, da so pogosto vidne naslednje napake: • napačen ton, ki zveni kot brošura ali oglas, • napačni časi/premalo različnih časov (preveč sedanjika in skoraj nikoli drugih), • SEO optimizirano izražanje (nenavadno v pesmih, vendar razumljivo, saj je SEO neželena pošta trenutno glavna uporaba generiranja AI besedila). Pozitivno pa lahko rečemo, da so rime dobre in dobro ustvarja, ko je tema abstraktna kot npr. občutek.
Ne izgledajo preveč kot pesmi, ampak bolj kot naključni stavki.
Ločila bi res lahko bila boljša.
Eden od njih bi se odlično prilegal v rap pesem.
Ne uporablja tem, ki se običajno uporabljajo v poeziji (ljubezen ...).
Običajno je slovnica precej v redu, razen nekaj napak, vendar je pomen pogosto nezanimiv in še bolj pomembno je, da je stopnja poetičnosti (ki je na koncu pravi kriterij) večino časa blizu ničle.

Nekatere besede, kot je blog itd., tako ali tako nikoli ne bi bile uporabljene v pesmi, zato bi jih bilo dobro odstraniti iz zbirke besed. Poleg tega pri nekem nenavadnem rimanem paru prva beseda para izsili drugo, da izstopi ne glede na pomen (opazil sem to pri besedi "trône").

Tabela 6: Predlogi francoskih posvetil

Kot lahko razberemo iz predlogov anketirancev, so se strinjali glede enakih problemov, ki jih ima generator. Ob tem pa lahko izpostavimo komentar, kjer je oseba govorila o nezaželeni pošti. V takih poštah so običajno podobne slovnične napake vse generirane na isti način. Tukaj se potem bolj nagibamo k RNN, ki proizvaja vnaprej določeno vsebino, zato ni inovativnosti in raznolikosti, ki ju imamo v poeziji.

Poleg tega nam vključitev besed, ki po navadi niso vključene v poeziji, lahko dodajo novo razsežnost ali idejo, kako bi se lahko poezija preoblikovala v prihodnosti. Tako bi generator lahko služil vsem pesnikom in ustvarjalcem, ter mogoče celo pisateljem. Vzeli bi dele ali celote posvetila in jih uporabili v svojih delih ali pa bi iz njih pridobili novo inspiracijo za ustvarjanje.

6 Zaključek

V diplomskem delu smo preizkusili kakovost generatorja posvetil v dveh jezikih. Implementirali smo nevronsko mrežo na osnovi modela GRU, ki je vseboval kodirnik in dekodirnik. Namen programa je bilo generiranje pesmi in posvetil ter ugotoviti, koliko kakovost upade z vključitvijo akrostiha. Poleg tega smo želeli ugotoviti, ali je za ustvarjanje poezije boljša francoščina ali angleščina.

Za implementacijo jezikovnih modelov smo uporabili prozaične članke in besedila, ki jih je vseboval korpus CommonCrawl. Za NMF in trigram model smo uporabili korpus z več kot 10 bilijard besed. Iz prvega smo nato pridobili 100 različnih tematik. Ob tem smo tudi določili točno določene rime, dolžine od štiri do osem vrstic.

Nato smo zbrali testno skupino ljudi, ki je prejela anketo, v kateri je bilo zbranih 10 posvetil v angleščini ali francoščini. Anketiranci so nato glede na gramatično pravilnost, pomen, poetičnost in splošno oceno podali svoje ocene od 1 do 5 (1 je najmanj, 5 največ). Poleg tega so ocenili, kako robotsko ali pa naravno se jim je zdelo vsako posvetilo. Na koncu so podali še svoje mnenje.

Iz rezultatov smo ugotovili, da uporaba akrostiha precej vpliva na kakovost, saj so bile primerjane naključne pesmi in pa najboljša posvetila, ki smo jih našli med testiranjem. Največja razlika se je pojavila v poetičnosti, kjer je kakovost padla za približno 5 %. Poleg tega so bili rezultati v francoščini slabši od angleških v vseh kriterijih, razen gramatiki. Če primerjamo rezultate glede na izdelavo pesmi, se pri angleščini vsi kriteriji poslabšajo za približno 4–5 %, medtem ko pri francoščini pomen in splošna ocena padeta za 20 %.

Sklepali smo, da je bil rezultat takšen, saj so bile teme naključne in niso vsebovale čustev ali misli, ki jih najdemo v vseh pesmih. Poleg tega lahko slabšo oceno poetičnosti pripišemo izbiri učne množice, saj je bila nevronska mreža naučena na prozaičnih besedilih. V kolikor bi vzeli za učne podatke samo poezijo, se le-ta precej razlikuje glede na literarna obdobja in na koncu bi dobili v posvetilu mešanico vseh. V prihodnje bi se dalo izboljšati generator, tako da bi pričel izdelavo novega posvetila, v kolikor predlagana vrstica ne bi bila med najboljšimi dvajsetimi izbirami. Metoda bi izboljšala kakovost in jo približala pesmim, ki vedno izberejo najboljšo vrstico glede na parametre. V nadaljnji raziskavi bi se vključilo več jezikov, ki bi jih primerjali. Tako bi lahko našli najboljši jezik za generacijo poezije in posvetil.

7 Literatura

[1] J. K. Brendan Bena. Introducing aspects of creativity in automatic poetry generation. 2020. *(Citirano na straneh VI in 13.)*

[2] E. L. Brid, Steven and E. Klein. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009. *(Citirano na straneh VII in 3.)*

[3] D. Britz. Recurrent neural networks tutorial. `https://dennybritz.com/posts/wildml/recurrent-neural-networks-tutorial-part-1/`, 2015. *(Citirano na straneh VII, 1, 6 in 7.)*

[4] J. R. M. Daniel W. Otter and J. K. Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems, vol. XX, no. X*, 2019. *(Citirano na strani 12.)*

[5] T. V. de Cruys. Automatic poetry generation from prosaic text. 2020. *(Citirano na straneh VII, 14, 15 in 17.)*

[6] R. Dey and F. M. Salem. Gate-variants of gated recurrent unit (gru) neural networks. 2017. *(Citirano na strani 9.)*

[7] A. Graves. Generating sequences with recurrent neural networks. 2014. *(Citirano na straneh VII in 8.)*

[8] K. Heafield. Kenlm: Faster and smaller language model queries. 2011. *(Citirano na strani 14.)*

[9] K. Heafield. Kenlm: Small and fast lm inference. `https://kheafield.com/code/kenlm/benchmark/`, 2011. *(Citirano na strani 14.)*

[10] J. Hopkins and D. Kiela. Automatically generating rhythmic verse with neural networks. 2017. *(Citirano na straneh VI in 11.)*

[11] G. H. Ilya Sutskever, James Martens. Generating text with recurrent neural networks. 2011. *(Citirano na strani 7.)*

[12] K. C. Junyoung Chung, Caglar Gulcehre and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014. *(Citirano na straneh VII in 9.)*

[13] M. Juršič. *Implementacija učinkovitega sistema za gradnjo, uporabo in evaluacijo lematizatorjev tipa RDR.* Bachelor's thesis, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, 2007. *(Citirano na strani 11.)*

[14] R. A. Kai-Ling Lo and P. Kurz. Gpoet-2: A gpt-2 based poem generator. 2022. *(Citirano na strani 12.)*

[15] A. Karpathy. The unreasonable effectiveness of recurrent neural networks. May 2015. *(Citirano na straneh VI, VII, 1, 2, 3, 4 in 5.)*

[16] D. Klein and C. D. Manning. Natural language grammar induction using a constituent-context model. 2002. *(Citirano na strani 10.)*

[17] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics. *(Citirano na strani 14.)*

[18] S. Kostadinov. Understanding gru networks. `https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be`, 2017. *(Citirano na strani 9.)*

[19] D. W. X. T. N. L. Z. T. Q. W.-Q. Z. Lanqing Xue, Kaitao Song and T.-Y. Liu. Deeprapper: Neural rap generation with rhyme and rhythm modeling. 2021. *(Citirano na strani 12.)*

[20] G. M. B. M. E. T. A. H. i. K. S. LOGAR BERGINC, Nataša. Korpusi slovenskega jezika gigafida. 2012. *(Citirano na strani 10.)*

[21] A. J. Nicole Gruber. Are gru cells more specific and lstm cells more sensitive in motive classification of text? 2020. *(Citirano na strani 9.)*

[22] C. Olah. Calculus on computational graphs: Backpropagation. `https://web.archive.org/web/20211113172142/http://colah.github.io/posts/2015-08-Understanding-LSTMs/`, 2015. *(Citirano na straneh VII in 8.)*

[23] C. Olah. Calculus on computational graphs: Backpropagation. `https://web.archive.org/web/20211114191011/http://colah.github.io/posts/2015-08-Backprop/`, 2015. *(Citirano na strani 7.)*

[24] K. K. Rajat Agarwal. Acrostic poem generation. 2020. *(Citirano na strani 13.)*

[25] A. Rosebrock. Softmax classifiers explained. https://pyimagesearch.com/2016/09/12/softmax-classifiers-explained/. *(Citirano na strani 6.)*

[26] G. L. Team. Natural language processing (nlp) tutorial: A step by step guide. https://www.mygreatlearning.com/blog/natural-language-processing-tutorial/#what-is-natural-language-processing, 2022. *(Citirano na straneh 10 in 11.)*

Priloge

A Anketa

Spoštovani! Sem študent računalništva in za diplomsko nalogo sem izdelal generator posvetil. Ta izpiše posvetilo, namenjeno kateri koli osebi. Generator posvetil lahko izpiše ogromno unikatnih posvetil, namenjenih določeni osebi. Da bi ugotovili kakovost posvetila v poetičnosti, pomenu in primerjavi glede na človeško literaturo, je treba izvesti anketo, kjer ljudje ocenijo več različnih posvetil in podajo svoje mnenje o izboljšavah.

Ker me v moji diplomski nalogi zanima, ali je mogoče, da bi lahko umetna inteligenca nadomestila človeško literaturo, sem se odločil, da naredim program, ki poda umetno narejena posvetila v angleščini in francoščini.

Prosil bi vas za sodelovanje.

Vzorec 1

Ocenite posvetilo glede na vsebino od 1 do 5:

> Pale lips, lips look crooked, from shiny hair to dark skin
> oh, i dont smoke, the boy replied
> emily took a deep breath and let the chill sink in
> then she closed the door and climbed inside

- gramatična pravilnost: 1 2 3 4 5
- pomen: 1 2 3 4 5
- poetičnost: 1 2 3 4 5
- splošna ocena: 1 2 3 4 5

Kako naravno se vam je zdelo posvetilo v angleščini?

- čisto robotsko
- moteče nenaravno
- nenaravno
- dober približek človeški literaturi
- ne bi ločil/a od človeške literature

Ocenite svoje znanje angleščine.

- Osnovno sporazumevanje
- Srednješolska raven – B1, B2
- Tekoče sporazumevanje

Vzorec 2

Ocenite posvetilo glede na vsebino od 1 do 5:

Parlons de la marque des fauteuils
optez pour une décoration simple et brève
envie de décorer vos meubles en un clin d' œil
mais c' est aussi l' occasion de réaliser votre rêve

- gramatična pravilnost: 1 2 3 4 5
- pomen: 1 2 3 4 5
- poetičnost: 1 2 3 4 5
- splošna ocena: 1 2 3 4 5

Kako naravno se vam je zdelo posvetilo v francoščini?

- čisto robotsko
- moteče nenaravno
- nenaravno
- dober približek človeški literaturi
- ne bi ločil/a od človeške literature

Ocenite svoje znanje francoščine.

- Osnovno sporazumevanje
- Srednješolska raven – B1, B2
- Tekoče sporazumevanje