Matematički fakultet, Univerzitet u Beogradu

Generisanje teksta - The Office

Autori:
Maša Mitrović, 359/2020
Nikola Radojičić, 110/2021

Sadržaj

1. Uvod

Generisanje teksta je proces automatskog stvaranja koherentnog i smislenog teksta, koji može biti u obliku rečenica, pasusa ili čak čitavih dokumenata. Ovaj proces obuhvata različite tehnike iz oblasti kao što su obrada prirodnog jezika (NLP), mašinsko učenje i algoritmi dubokog učenja, koje analiziraju ulazne podatke i generišu tekst sličan onom koji pišu ljudi. Cilj generisanja teksta je da proizvedeni sadržaj bude ne samo gramatički ispravan, već i kontekstualno odgovarajući i zanimljiv za ciljnu publiku.

Prednosti generisanja teksta:

- **Povećana efikasnost:** Generisanje teksta može značajno smanjiti vreme i trud potrebne za izradu velike količine tekstualnog sadržaja. Na primer, može se koristiti za automatizaciju kreiranja opisa proizvoda, objava na društvenim mrežama ili tehničke dokumentacije. Time se ne samo štedi vreme, već se timovima omogućava da se fokusiraju na strateški važnije zadatke.
- **Unapređena kreativnost:** Veštačka inteligencija može velikom brzinom da generiše jedinstven i originalan sadržaj, koji bi ljudima često bilo teško ili nemoguće da proizvedu ručno.
- **Povećana dostupnost:** Generisanje teksta može pomoći osobama sa invaliditetom ili onima koji se suočavaju sa jezičkim barijerama, kroz kreiranje sadržaja u alternativnim formatima ili na različitim jezicima. Na taj način informacije postaju dostupnije širem krugu ljudi, uključujući gluve i nagluve osobe, govornike kojima je strani jezik drugi jezik, kao i osobe sa oštećenim vidom.
- **Bolja interakcija sa korisnicima:** Personalizovano i prilagođeno generisanje teksta omogućava kompanijama da ostvare kvalitetniju komunikaciju sa svojim korisnicima. Prilagođavanjem sadržaja korisniku moguće je stvoriti relevantnije i smislenije interakcije, što dovodi do većeg zadovoljstva i lojalnosti korisnika.
- **Unapređeno učenje jezika:** Generisanje teksta može biti koristan alat za učenje jezika, jer omogućava davanje povratnih informacija i predloga za unapređenje. Generisanjem teksta u određenom stilu ili žanru, učenici mogu vežbati i razvijati svoje veštine pisanja na strukturisaniji i vođen način.

2. O skupu podataka

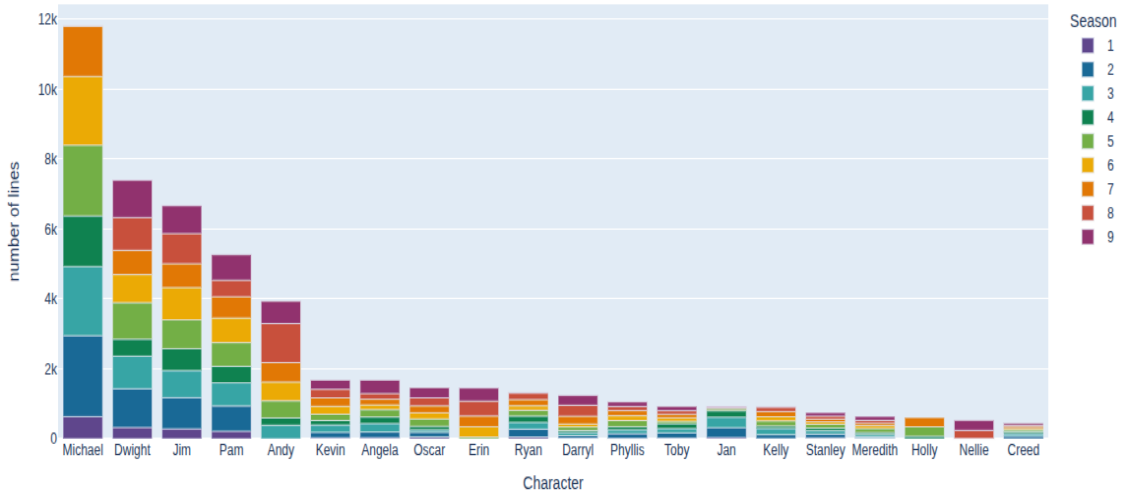
Skup podataka koji se koristi u ovom projektu sastoji se od citata iz američke TV serije *U kancelariji* (eng. *The Office*) i obuhvata svaku repliku iz svih sezona serije. Skup sadrži 58721 instancu koje predstavljaju 58721 replika izrečenih u seriji. Svaka instanca u skupu podataka sadrži jedinstveni indeks, ime lika koji izgovara repliku, sam tekst replike, kao i informacije o sezoni i broju epizode u kojoj se replika pojavljuje. Ovakva struktura skupa podataka čini ga pogodnim za primenu tehnika obrade prirodnog jezika i generisanja teksta, jer omogućava učenje jezičkih obrazaca karakterističnih za pojedinačne likove i različite delove serije.

Cilj ovog projekta je primena tehnika generisanja teksta kako bi se kreirao model sposoban da proizvodi nove replike u stilu likova iz serije *The Office*. Korišćenjem postojećeg skupa podataka koji sadrži dijaloge iz svih sezona serije, model se trenira da prepozna jezičke obrasce, stil govora i karakteristične izraze pojedinačnih likova. Na taj način, projekat ima za cilj da demonstrira kako se metode obrade prirodnog jezika i mašinskog učenja mogu koristiti za generisanje koherentnog i smislenog teksta, kao i da se prikažu mogućnosti i ograničenja savremenih modela generativne veštačke inteligencije u praktičnoj primeni.

Slika 1: The Office

3. Pretprocesiranje podataka

U procesu obrade podataka izvršeno je nekoliko koraka kako bi se skup prilagodio ciljevima projekta i poboljšao kvalitet treniranja modela. Najpre je analizirana učestalost pojavljivanja likova u seriji i izdvojeno je 20 likova sa najvećim brojem replika, nakon čega su zadržane isključivo njihove replike. Iako skup podataka sadrži oko 700 različitih likova, većina njih ima veoma mali broj replika, zbog čega nisu relevantni za učenje modela. Replike glavnih likova čine 50684 od ukupno 58721 zapisa, što iznosi približno **86,3%** celokupnog skupa podataka.



Slika 2: Grafik sa brojem replika

Nakon toga, skup je dodatno filtriran tako što su uklonjene replike koje se sastoje od samo jedne reči, kao i replike duže od 25 reči, jer kratki one-lineri i predugački dijalozi nisu pogodni za generisanje kvalitetnog teksta. Nakon svih koraka procesiranja, konačan skup podataka sadrži 39917 replika koje se koriste za dalju analizu i treniranje modela.

Podatke smo podelili koristeći funkciju *train_test_split* u odnosu 80:20, gde je veći deo namenjen **treniranju** modela da prepozna stil i strukturu dijaloga, dok je preostalih 20% izdvojeno za **testiranje**. Ova podela je neophodna kako bismo objektivno ocenili model na podacima koje ranije nije video i osigurali da on zaista uči obrasce, a ne da ih pamti napamet, odnosno da ne dolazi do preprilagođavanja.

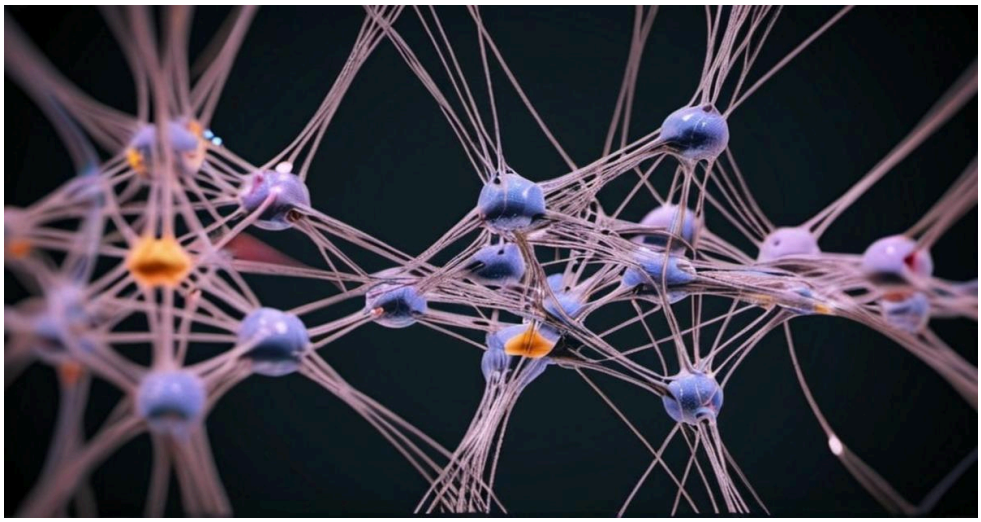
4. Kreiranje modela

Postoje različite tehnike generisanja teksta koje se koriste u oblasti obrade prirodnog jezika. Među ovim tehnikama najpoznatiji su N-gram modeli, CRF, RNN, LSTM, GRU i Transformer arhitektura. Ove tehnike obuhvataju kako klasične statističke pristupe, tako i savremene metode zasnovane na neuronskim mrežama, koje omogućavaju automatsko stvaranje koherentnog i smislenog tekstualnog sadržaja.

4.1. Ukratko o neuronskim mrežama

Neuronske mreže oponašaju način na koji ljudski mozak obraduje informacije, koristeći veštačke neurone koji obrađuju podatke kroz slojeve. Svaki neuron prima ulazne podatke (npr. podatke iz okruženja ili drugih neurona), množi ih odgovarajućim težinama (w1, w2, w3..) i primenjuje aktivacionu funkciju kako bi odlučio da li će poslati signal i sa kojom snagom.

Za generisanje teksta ranije su se najviše koristile **RNN** i **LSTM** mreže. One su obrađivale reč po reč, s leva na desno. Problem je bio u tome što bi "zaboravile" početak duge rečenice dok stignu do kraja. Danas dominiraju **Transformatori.** Njihova ključna inovacija je **mehanizam pažnje (self-attention)**. Umesto da čitaju linearno, oni vide celu rečenicu odjednom i matematički izračunavaju koja je reč najbitnije povezana sa kojom, bez obzira na njihovu udaljenost u tekstu.



Slika 3: Neuronske mreže

4.2. LSTM

Osnovu za rad sa sekvencijalnim podacima čine rekurentne neuronske mreže (RNN), koje su specifične po tome što poseduju povratne veze koje im omogućavaju prenos informacija kroz vremenske korake. Međutim, **LSTM (Long Short-Term Memory)** mreže predstavljaju napredniji tip ove arhitekture, namenski razvijen kako bi se prevazišli nedostaci bazičnih RNN modela. Ključna prednost LSTM-a ogleda se u sposobnosti efikasnog upravljanja informacijama na dužim vremenskim intervalima, što se postiže upotrebom memorijskih ćelija i specifičnog sistema "vrata" (ulazna, izlazna i vrata zaborava). Zahvaljujući ovakvoj strukturi, uspešno se rešava problem nestajanja gradijenta, što omogućava stabilno modelovanje dugoročnih zavisnosti unutar teksta.

U okviru ovog rada, LSTM model je odabran kao optimalno rešenje za generisanje teksta zbog svoje efikasnosti u obradi nizova reči i prepoznavanju konteksta unutar dijaloga. Imajući u vidu da replike u seriji *The Office* poseduju specifičnu strukturu i karakterističan humor, primena LSTM mreže omogućava modelu da identifikuje obrasce u govoru likova i uspostavi logičku povezanost između reči. Na taj način, generisane replike zadržavaju koherentnost i stilsku usklađenost sa originalnim podacima, što direktno doprinosi realizaciji postavljenih ciljeva projekta.

4.3. Word-level

Word-level LSTM model predstavlja specifičan pristup u obradi prirodnog jezika (NLP) gde se kao osnovna jedinica (token) uzima cela reč, a ne pojedinačni karakter. Primetna razlika u odnosu na karakterne modele je to što model lakše prati smisao (semantiku) i gramatičku ispravnost (sintaksu) rečenice, jer radi sa gotovim rečima koje treba da postavi na pravo mesto.

4.3.1 Princip rada

Kako bi mogli da kreiramo model, potrebno je da koristimo klasu **Tokenizer**, pomoću koje transformišemo podatke iz serije u nizove celih brojeva. Svaka jedinstvena reč dobija svoj jedinstveni indeks. Način funkcionisanja modela je po principu "**Many-to-one"** arhitekture:

- **Ulaz**: Niz od N prethodnih reči.
- **Izlaz:** Predviđanje ($N+1$) reči na osnovu prethodnih.

4.3.2 Arhitektura slojeva

Struktura ove mreže je podeljena u nekoliko nivoa da bi se dobio što kvalitetniji sadržaj:

- **Embedding sloj:** Vektorski prostor dimenzije 150 u koji se mapiraju reči predstavljene brojevima. Pomoću ovog sloja model uči relacije između reči.
- **LSTM slojevi:** Koriste se dva sloja (sa 256 i 128 jedinica) koji čine srž modela. Korišćenjem mehanizma vrata (gates), ovi slojevi selektuju relevantne informacije iz dijaloga, što je presudno za učenje specifičnog humora i interakcije likova u seriji.
- **Dropout slojevi (0.3 i 0.4):** Ova tehnika nasumično isključuje određeni procenat (30% i 40%) neurona tokom treninga. Ovim se sprečava ***overfitting*** (preprilagođavanje), osiguravajući da model ne nauči replike napamet, već da uspešno generalizuje stil govora.

```
opt = Adam(learning_rate=0.0005)

model = Sequential([
    Embedding(total_words, 150, input_length=max_sequence_len-1),
    LSTM(256, return_sequences=True),
    Dropout(0.4),
    LSTM(128),
    Dropout(0.3),
    Dense(total_words, activation='softmax')
], name="Triple_LSTM_v1_256-128")

model.compile(loss='sparse_categorical_crossentropy', optimizer=opt, metrics=['accuracy', perplexity])
model.summary()
```

Slika 4: LSTM Word-level model

4.3.3 Evaluacija modela

Uspešnost modela je merena pomoću metrika **tačnosti (accuracy)** i **perplexity.** Dok tačnost meri procenat pogodaka, perplexity meri nivo nesigurnosti (zbunjenosti) modela pri predviđanju sledeće reči te se prvenstveno koristi za evaluaciju jezičkih modela. Jednostavnije rečeno, niži rezultat ukazuje na to da je model “manje zbunjen”, što ga čini boljim u predviđanju sledeće reči.

4.4. Char-level

Nasuprot modelu baziranom na rečima, **Character-level LSTM** posmatra tekst kao niz pojedinačnih karaktera (slova, znakova interpunkcije i razmaka). Ovakav pristup omogućava modelu da nauči samu strukturu pisanja, pravopis i specifičnu interpunkciju serije, te je znatno fleksibilniji kada je u pitanju rečnik, jer nije ograničen unapred definisanim skupom reči.

4.4.1 Princip rada

Prvi korak u razvoju modela obuhvata prečišćavanje korpusa (normalizaciju) i svođenje teksta na mala slova uz zadržavanje samo neophodnih znakova interpunkcije. Za razliku od reči, ovde se kreira rečnik jedinstvenih karaktera, gde se svakom karakteru dodeljuje jedinstveni numerički indeks. Za generisanje trening primera korišćena je tehnika klizećeg prozora (**sliding window**):

- **max_sequence_len (30):** Model posmatra sekvencu od 30 uzastopnih karaktera.
- **step (2):** Prozor se pomera za dva karaktera unapred, čime se definiše gustina uzorkovanja podataka.
- **Cilj:** Na 30 karaktera, model uči da predvidi 31. karakter u nizu.

4.3.2 Arhitektura slojeva

Model je definisan kao sekvencijalni niz slojeva optimizovanih za prepoznavanje mikro-obrazaca u tekstu:

- **Embedding sloj:** Svaki karakter (predstavljen indeksom) mapira se u vektorski prostor dimenzije 50. Iako je prostor manji nego kod *word-level* modela, on je dovoljan da model nauči osnovne relacije između karaktera (npr. koji karakteri često idu jedan uz drugi).
- **LSTM slojevi:** Arhitektura se sastoji od dva sloja sa 128 i 64 jedinice. Prvi sloj vraća sekvence, što omogućava drugom sloju da analizira dublje vremenske zavisnosti unutar prozora od 30 karaktera.
- **Dropout sloj (0.2):** Primenjena je blaga regularizacija kako bi se osigurala bolja generalizacija i sprečilo doslovno kopiranje delova scenarija.
- **Dense sloj sa softmax aktivacijom:** Izlazni sloj ima onoliko neurona koliko ima jedinstvenih karaktera u rečniku. *Softmax* funkcija pretvara

izlaz u distribuciju verovatnoća, dajući nam procenu koji karakter najverovatnije sledi.

Za funkciju gubitka izabrana je **Sparse Categorical Crossentropy**, koja je idealna za klasifikacione probleme gde su ciljne vrednosti celi brojevi (indeksi karaktera). Optimizacija se vrši putem **Adam** algoritma, koji se dinamički prilagođava stopi učenja tokom treninga, dok se uspešnost modela prati kroz metriku **tačnosti** (*accuracy*) tokom svake epohe.

```
embedding_dim = 50

lstm_units_1 = 128
lstm_units_2 = 64

model = Sequential([
    Embedding(input_dim=vocab_size, output_dim=embedding_dim),
    LSTM(lstm_units_1, return_sequences=True),
    Dropout(0.2),
    LSTM(lstm_units_2),
    Dropout(0.2),
    Dense(vocab_size, activation='softmax')
])

model.compile(
    loss='sparse_categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
```

Slika 5: Char-level model

4.5. Generisanje teksta

Nakon što se učenje završi, modeli (i na nivou reči i na nivou karaktera) ne biraju sledeći simbol prosto tako što uzmu onaj sa najvećom šansom. Umesto toga, koristi se funkcija za uzorkovanje koja nam omogućava da biramo sledeći token na osnovu verovatnoća koje je model izračunao. Glavna stvar u tom procesu je parametar **temperature (T)**, koji nam služi da kontrolišemo koliko će model biti „hrabar“ ili kreativan u svom pisanju.

Uticaj temperature na izlazne rezultate izgeda ovako:

- **Niske vrednosti temperature (T < 0.6):** Model postaje konzervativniji i fokusira se isključivo na tokene (reči/karakteri) sa najvećom verovatnoćom. Rezultat je tekst koji je gramatički tačan i stabilan, ali

često zna da se ponavlja i koristi stalno iste, predvidive fraze koje se najčešće pojavljuju u seriji.

- **Visoke vrednosti temperature (T > 1.0):** Model postaje kreativniji jer temperatura ujednačava šanse za sve tokene. To znači da će model češće birati neke neobične ili ređe reči, što dijaloge čini zanimljivijim i duhovitijim. Ipak, ako se pretera, rečenice mogu postati potpuno besmislene i izgubiti logiku.

Podešavanjem temperature zapravo tražimo zlatnu sredinu - da replike zvuče kao da su iz serije, ali da ne budu samo prepisani delovi postojećeg scenarija.

5. Procene kvaliteta modela

Prilikom upoređivanja rezultata, primetno je da model na nivou karaktera (**Character-level**) postiže znatno veću vrednost tačnosti u poređenju sa modelom na nivou reči (**Word-level**). Iako na prvi pogled to može delovati kao da je model na nivou karaktera uspešniji, važno je razumeti prirodu samog zadatka.

Razlog za ovu razliku leži u složenosti predviđanja:

- **Kod karakternog modela**, model bira jedan od 32 moguća simbola. Recimo, mnogo je lakše statistički pogoditi da posle slova „n“ u engleskom jeziku često ide „o“ ili „e“, nego pogoditi celu reč.
- **Kod modela na nivou reči**, model mora da odabere jednu reč iz rečnika koji sadrži hiljade unikatnih pojmova. Zbog toga je verovatnoća direktnog pogotka znatno manja, što rezultira nižim *accuracy* vrednostima.

Stoga, veća tačnost kod karakternog modela ne znači nužno da je on bolji. Naprotiv, word-level model, uprkos manjoj tačnosti, često generiše smislenije rečenice jer operiše čitavim konceptima i rečima, dok se karakterni model bavi mikroskopskom strukturom teksta, pa se ponekad dešava da generiše nepostojeće reči.

```
seed = "thats what she"
print("CHAR-LEVEL: " + generate_single_line(char_model, seed, char_to_index, index_to_char, length=200, temperature=0.4))
print("WORD-LEVEL: " + generate_text_with_temp(seed, 15, word_model, max_sequence_len, temperature=0.4))
```

```
CHAR-LEVEL: thats what she say it.
WORD-LEVEL: thats what she was a very good idea i don't know what you have a little name of
```

```
seed = "thats what she"
print("CHAR-LEVEL: " + generate_single_line(char_model, seed, char_to_index, index_to_char, length=200, temperature=1))
print("WORD-LEVEL: " + generate_text_with_temp(seed, 15, word_model, max_sequence_len, temperature=1))
```

```
CHAR-LEVEL: thats what she need connandly. my firtin, and thats but hes indegilf
WORD-LEVEL: thats what she water lost the office corporate do really would've will just leave her in a grill
```

Slika 5: Primer generisanja teksta

5.1. Poređenje modela sa GPT-2 modelom

U okviru ovog poglavlja izvršeno je poređenje predloženog Word-level LSTM modela sa pre-trained GPT-2 modelom (HScomcom/gpt2-theoffice) kako bi se procenila efikasnost sopstvenog rešenja u odnosu na savremenu arhitekturu zasnovanu na Transformerima.

```
generate_comparison("MICHAEL: Dwight, you are")
generate_comparison("JIM: (to camera) Yesterday,")
```

```
--- SEED: MICHAEL: Dwight, you are ---
LSTM: MICHAEL: Dwight, you are going to be a real good time and i got to spend the way so
GPT-2:  MICHAEL: Dwight, you are a suspect.
--------------------------------------------------
--- SEED: JIM: (to camera) Yesterday, ---
LSTM: JIM: (to camera) Yesterday, we're gonna be here of the regional phone i don't know you know what you
GPT-2:  JIM: (to camera) Yesterday, I was just back from my first day at work, so I figured I'd catch you walking to
--------------------------------------------------
```

Sintaksička ispravnost:

- LSTM: Model pokazuje sposobnost generisanja fraza koje podsećaju na seriju, ali često gubi gramatičku logiku u drugom delu rečenice. Ovo je posledica ograničenog kapaciteta LSTM ćelija da zadrže dugoročne zavisnosti.
- GPT-2: Generiše potpuno ispravne i prirodne rečenice na engleskom jeziku.

Kontekstualno razumevanje:

- LSTM: Uspešno identifikuje vokabular, ali ga koristi na konfuzan način.
- GPT-2: Pokazuje dublje razumevanje situacije i vidnu prednost Self-Attention mehanizma u razumevanju tona karaktera.

Naš LSTM model je treniran od nule na specifičnom skupu podataka sa značajno manje parametara. GPT-2 model koristi znanje stečeno na milijardama reči, što mu daje prednost u poznavanju samog jezika. LSTM model je pokazao da može uspešno da nauči "rečnik" serije, ali za složene narativne strukture, Transformer arhitektura je značajno bolja.

6. Zaključak

U ovom radu implementirani su **Word-level** i **Character-level LSTM** modeli za generisanje replika iz serije *The Office*. Testiranjem i analizom oba modela, došli smo do zaključka da **Word-level model** pruža bolje rezultate, jer lakše održava smisao i gramatičku strukturu rečenica, dok model na nivou karaktera zahteva više vremena i povremeno daje rezultate koji nisu gramatički ispravni.

Iako su postignuti rezultati zadovoljavajući, model ima svoja ograničenja i nije savršen. Danas postoje bolje arhitekture od običnih LSTM modela koje se češće koriste radi generisanja teksta. Da bismo dobili bolje rezultate, potrebno je imati bolje resurse za rad, manja vremenska i memorijska ograničenja i znatno veći skup podataka.

Reference

[1] Predrag Janičić, Mladen Nikolić: Veštačka inteligencija, 2024.

[2] Keras API docs.

[3] Dan Jurafsky, James H. Martin: Speech and Language Processing, 2026.

[4] Materijali s kurseva Računarska inteligencija i Istraživanje podataka 1