UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET
Katedra za računarstvo

Generisanje opisa fotografija

- Duboko učenje -

Mentor: prof. dr Aleksandar Milosavljević

Studenti: Natalija Stamenković 1258
Milica Todorović 1256

Niš, 2021.

Sadržaj

Uvod

Generisanjem opisa slika se dobija jasan I precizan opis fotografije koja se prosleđuje na ulazu. Rešavanje problema uključuje tehnike računarskog vida (*eng. Computer vision*) za ekstrakciju fičera sa fotografije kao I tehnike obrade prirodnih jezika (*eng. Natural language processing)* da bi se generisao tekstualni opis date fotografije.

Za generisanje opisa slika koriste se arhitekture zasnovane na *enkoder - dekoder* rekurentnim neuronskim mrežama koje su se pokazale veoma efikasnim u rešavanju ovog problema. Implementacija ovih arhitektura se može podeliti na *injected* bazirane modele I *merged* bazirane modele. Razlika između prethodno navedenih modela je u načinu korišćenja RNN mreža u njihovoj implementaciji.

Postoji mnogo načina za implementaciju rešenja ovog problema. Zajedničko za sva rešenja je korišćenje pre-treniranih CNN modela za enkodiranje slike. Pre-trenirani model se učitava, pri čemu se uklanja njegov izlaz, zatim se propuštanjem slike kroz dati model dobija njena interna reprezentacija koja se koristi kao enkodirana slika.

Arhitektura mreže

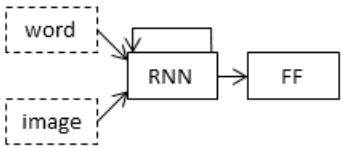
Kao što je prethodno napomenuto, arhitekture korišćnjene za rešavanje datog problema baziraju se na *enkoder – dekoder* arhitekturama. Ovakve arhitekture uključuju dva glavna elementa:

- *Enkoder* – mrežni model koji detektuje fotografiju sa ulaza I enkodira sadržaj u vektor fiksne dužine korišćenjem interne reprezentacije.
- *Dekoder* – mrežni model koji kao ulaz koristi enkodiranu fotografiju I generiše tekstualni opis te fotografije.

Generalno konvolucione neuronske mreže se koriste za enkodiranje fotografije. Dok se rekurentne neuronske mreže (RNN), konkretno LSTM (*Long-Short Term Memory*), koriste za enkodiranje tekstualne sekvence i/li za generisanje sledeće reči u sekvenci.

Injected model

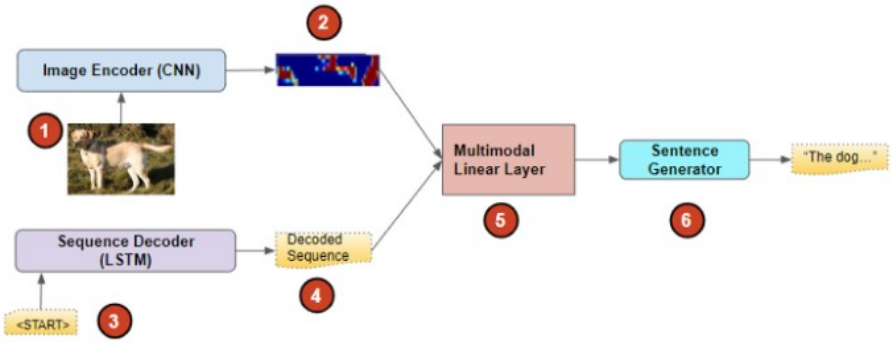
Inject-ovani model kombinuje enkodiranu formu slike sa svakom reči iz sekvence teksutalnog opisa slike generisanog do tada. Ovaj model koristi RNN model u kome fičeri slika direktno učestvuju u okviru rekurentne mreže tokom procesa enkodiranja sekvence.



Slika 1: Injektovanje slike u isti RNN koji procesira reči.

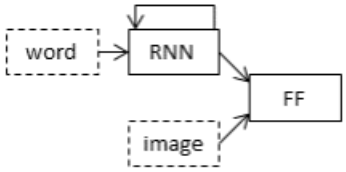
Merged model

Ovaj model funkcioniše tako što razdvaja enkodiranje lingvističkih delova od enkodiranja fičera slike i kasnije ih spaja u zasebnom sloju. U ovom tipu model RNN funkioniše tako što primarno enkodira reči, dok kasnije to spaja sa ekstraktovanim fičerima fotografija.



Slika 2: Merged model

Merged arhitektura tretira RNN ekskluzivno kao jezički model za enkodiranje lingvističkih sekvenci. Lingvistički vektori, zajedno sa fičerima fotografija, se koriste u multimodalnom sloju koji može biti *linearni* ili *softmax* sloj. Primarno RNN je enkoder lingvističkih informacija. Ovaj model omogućava kombinaciju različitih reprezentacija slika sa konačnim RNN slojem pre predikcije.

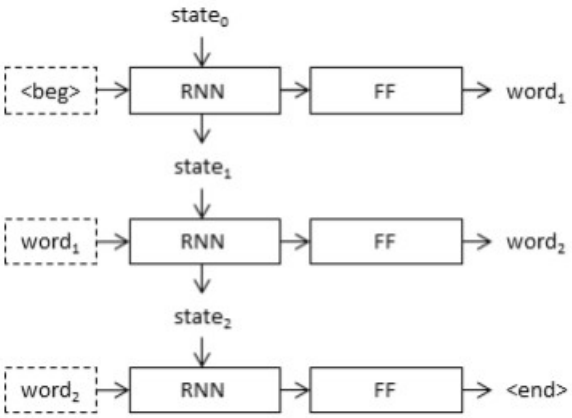


Slika 3: Spajanje slike sa ouput-om RNN - a posle procesiranja reči.

Komponente arhitekture mreže

RNN i predviđanje sekvenci

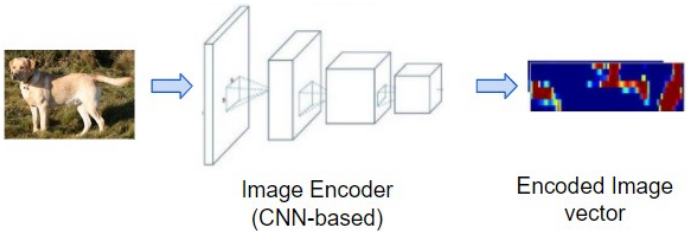
U procesiranju jezika, RNN enkodira prefiks (napr. opis generisan do sada) i sam predviđa narednu reč u sekvenci (uz pomoć *feed forward* sloja) ili prosleđuje enkodiranje sledećem sloju koji je izvršiti predikciju. Ova naredna reč dodaje se prefiksu na sledećoj iteraciji kako bi se predvidela sledeća reč, sve dok se ne dođe do simbola za kraj sekvence. Uglavnom, predikcija se vrši korišćenjem softmax funkcije.



Slika 4: Proces predviđanja sekvence

LSTM, je tip RNN koji se koristi za predviđanje sekvenci. Bazirano na prethodnom tekstu možemo predvideti koja će sledeća reč biti. Dokazano da je ovaj tip mreže efektivniji od tradicionalnih RNN. LSTM može da zadrži relevantne informacije kroz celokupno procesiranje ulaza, dok korišćenjem „kapije zaboravljanja“ (*eng. forget gate*) odbacuje nerelevatne informacije. Treba napomenuti da je moguće koristiti i druge tipove RNN, umesto prethodno navedene LSTM, ali će se ovaj rad, kao i data implemetacija zadržati na LSTM.

Enkoder fičera fotografija



Slika 5: Enkoder - CNN

Ovaj deo generisanja opisa slika uzima izvornu sliku i kao izlaz generiše enkodiranu reprezentaciju te slike koja obuhvata njene bitne karakeristike.

Kao enkoder, uglavnom se koristi pret-renirani CNN model za klasifikaciju slike, a zatim se uklanja finalni sloj, koji je zapravo klasifikator. Moguće je koristiti različite CNN mreže u ovom koraku, pri čemu se ovaj rad ograničava na *VGGNet* i *Inception*.

Model konvolucione mreže sastoji iz određenog broja CNN blokova koji progresivno ekstraktuju različite fičere sa fotografija i generišu odgovarajaću mapu fičera, koja obuhvata najbitnije elemente jedne fotografije.

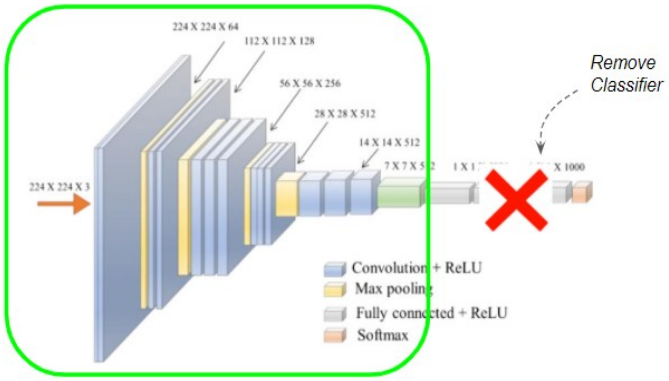
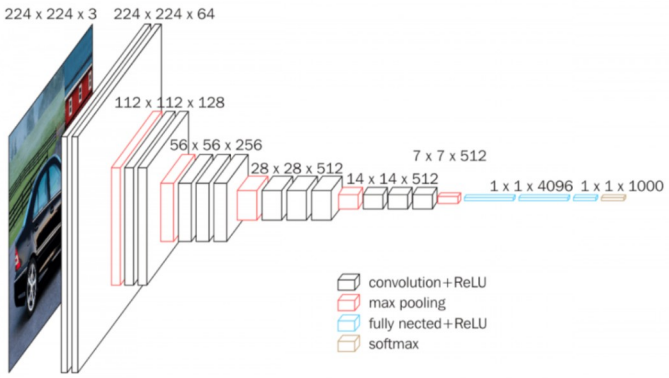


Figure 6: Korišćenje CNN mreže

Mreža počinje izdavajanjem jednostavnih geometrijskih oblika kao što su krivine i polu-krugovi a zatim nastavlja do složeniijih elemenata kao što nosevi, oči, ruke i eventualno prepoznaje elemente kao što su točkovi, lice. itd. Izvorno ovaj model kao rezultat identifikuje oblike sa slike, međutim, kada se primenjuje na generisanje opisa slika, uklanja se poslednji sloj, i kao izlaz uzimaju se samo ekstraktovani fičeri sa fotografija.

VGG-16

VGG-16 je konvoluciona neuronska mreža koja je duboka 16 slojeva. Model koristi skup težina treniranih na ImageNet skupu podataka.

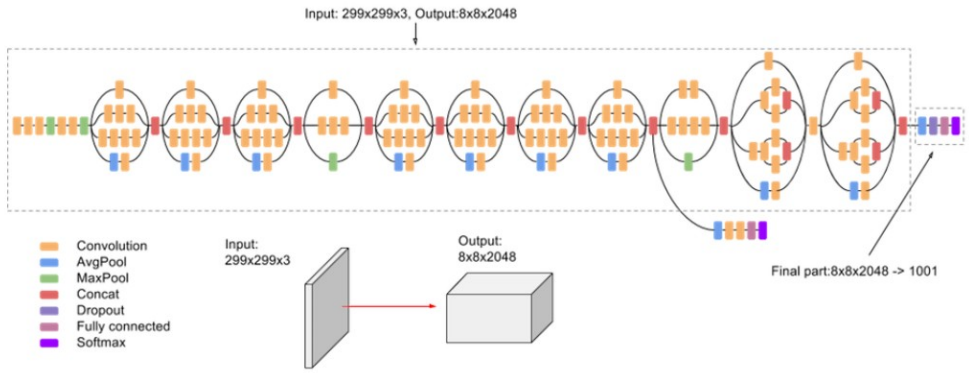


Slika 7: VGG-16 mreža

VGG-16 sadrži samo konvolucione i *pooling* slojeve. Koristi samo kernele dimenzija 3x3 za konvoluciju. Dimenzije za *max-pooling* su 2x2 za sve slojeve u modelu. Ima oko 138 miliona paramtera. Treniran je na *ImageNet* podacima sa preciznošću 92.7%. Takođe postoji verzija *VGG-19* sa 19 slojeva. Ulaz za *VGG-16* je 224x224 piksela sa tri kanala za RGB.

InceptionV3

InceptionV3 je široko korišćen model za prepoznavanje objekata sa slike sa preciznošću 78.1%. Model sadrži asimetrične blokove, koji uključuju konvolucione, *average*, *pooling*, *max-pooling*, konkatenacione, *dropout* kao I potpuno povezane slojeve. Dimenzija ulaza je 229x229 piksela uz tri kanala za RGB.



Slika 8: InceptionV3 mreža

Implementacija

Uz ovaj rad data je implementacija opisanih metoda. Implementacija je izvršena u Python programskom jeziku, a dostupna je kao Jupiter Notebook na Google Colab-u. Najznačajnije korišćene biblioteke Tensorflow (Keras) i NLTK. U nastavku ovog odeljka biće dat opis skupa podataka, korišćenih metrika za evaluaciju, kao i opis dobijenih rezultata.

Skup podataka

Ovaj projekat koristi Flicker8k skup podataka, dostupan na: https://www.kaggle.com/ming666/flicker8k-dataset. Skup podataka sastoji se iz ~ 8000 slika, pri čemu svaka ima unikatno ime. Pored slika, skup podataka sadrži i opise datih slika, pri čemu svaka slika može imati više od jednog opisa, za denfikaciju opisa (kojoj slici pripada) koristi se naziv odgovarajuće slike kao njen identifikator (*Flickr_8k.token.txt* fajl). Takođe, fajlovi *Flickr_8k.trainImages.txt* i *Flickr_8k.devImages.txt* sadrže identifikatore slika koje se uzimaju respektivno za treniranje i testiranje modela.

Preprocesiranje

Kao što je prethodno napomenuto, korišćena arhitektura podrazumeva ekstrakciju fičera korišćenjem neke pre-trenirane mreže. Kako bi se uštedelo na vremenu izvršenja, deo preprocesiranja svodi se na ekstraktovanje fičer vektora iz trening i test slika a zatim čuvanje dobijenih vektora u okviru fajlova. Dati fajlovi se, prilikom treniranja, mogu samo učitati, umesto da se ekstrakcija vrši pri svakoj trening sesiji (*extract_features()* metoda). Kao što je prethodno napomenuto, ova implementacija podrazumeva korišćenje VGG-16 i InceptionV3 mreža, te su one iskorišćene u ovom koraku.

Pored toga, neophodno je očistiti opise slika date u skupu podataka. Pripremanje opisa slika podrazumeva klasično preprocesiranje prirodnog jezika, naime, uklanjanje interpunkcije, alfa-numeričkih znakova, prebacivanje u sva mala slova itd (*clean_captions()* metoda).

Priprema podataka za treniranje modela

Ova implementacija podrazumeva kreiranje sekvenci reči od datih opisa slika, što je delimično omogućeno korišćenjem *Tokenizer* klase *Keras* biblioteke. Data klasa omogućava vektorizaciju korpusa teksta, tako što pretvara svaki tekst u sekvencu celih brojeva, pri čemu svaki celobrojni broj predstavlja indeks odgovarajućeg tokena u *dictionary*-ju. Istrenirani tokenizator koristi se u okviru *create_sequences()* metode, koja za svaki opis slike izdvaja ulaznu sekvencu reči i izlaznu reč (enkodiranu kao kategorička vrednost).

S obzirom da se radi sa većom količinom slika i opisa, učitavanje svih potrebnih informacija odjednom zahteva veću količinu RAM memorije, te je potrebno definisati custom *data_generator* funkciju, koja će podatke učitavati u *batch*-evima određene veličine.

Korišćeni modeli

U prethodnom poglavlju navedene su arhitekture mreža koje se mogu koristiti za rešavanje ovog problema. Implementacija data uz ovaj rad, ograničava se na prethodno opisanu *merged*

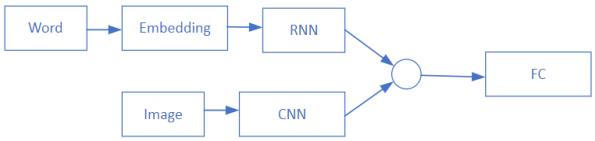
arhitekturu, s obzirom da se ona prihvata kao superiornija u odnosu na *injected* model. Isprobane su dve varijante ove arhitekture, jedna koja u svom Embedding sloju koristi GloVe model i druga koja ga ne koristi. Obe arhitekture koriste *Adam* optimizator i kategoričku kros-entropiju kao funkciju gubitka.

Prilikom treniranja modela, najbolji model sačuvan je u okviru odgovarajućeg fajla, korišćenjem *ModelCheckpoint* klase. Takođe, omogućeno je i čuvanje *log*-a treniranja, za kasniju vizuelizaciju procesa treniranja.

Proces treniranja podrazumeva eksperimentisanje sa korišćenjem određenog modela CNN mreže, promena u Embedding sloju, procentom Dropout-a i veličinom Dense i LSTM slojeva (broj neurona).

GloVe

Glove je *word embedding* model. Predstavlja nenadgledani mehanizam za dobijanje reprezentacije reči. Ovo se postiže mapiranjem reči u smisleni prostor gde je distanca između reči povezana sa semantičkom sličnošću između reči. Relacije između reči se čuvaju u matrici.



Slika 9: GloVe model

U ovom radu, u jednom od modela, kao što je napomenuto, korišćen je pre-trenirani GloVe model za reprezentaciju reči.

Evaluacija modela

Nakon treniranja navedenih varijacija modela sa različitim hiperparametrima, na osnovu BLEU skora, za svaki model, određeni su hiperparametri koji maksimizuju njegovu uspešnost. Finalna evalucija i poređenje svih modela izvršena je na osnovu objektivne BLEU metrike, kao i subjektivne procene valjanosti generisanih opisa nad „novim“ slikama.

BLEU skor (*Bilingual Evaluation Understudy)*

BLEU je skor poređenja prevoda teksta sa jednim ili više referentnim prevodom. I ako je razvijen zbog prevođenja, može se koristiti za evaluaciju teksta generisanog u NLP taskovima. BLEU je metrika evaluiranja generisanje rečenice u poređenju sa referentnom rečenicom.

Implementacija ovog skora, može se naći u okviru NLTK biblioteke. U ovom radu je korišćenja funkcija *corpus_bleu()* za računanje BLEU skora za više rečenica ili paragrafa dokumenta. Referentne rečenice moraju biti specificirane kao lista dokumenta gde je svaki dokument lista referenci i svaka alternativna referenca je lista tokena.

Primer definisanja referenci:

```
# two references for one document
from nltk.translate.bleu_score import corpus_bleu
references = [[['this', 'is', 'a', 'test'],
['this', 'is' 'test']]]
candidates = [['this', 'is', 'a', 'test']]
score = corpus_bleu(references, candidates)
print(score)
```

Kumulativni i individualni BLEU skorovi

Računanje ovog skora korišćenjem NLTK biblioteke dozvoljava specificiranje težina različitih n-grama. To daje fleksiblinost za računanje različitih tipova BLEU skorova, kao što su *individualni* i *komulativni* n-gram skorovi:

- **Individualni n-gram skorovi**: Evaluacija poklapanja grama u specifičnom redosledu, kao što je jedna reč (1-gram) ili par reči (2-gram) itd. Težine su specificane kao torka gde svaki indeks referencira redosled grama. Da bi se izračunao BLEU skor samo za 1-gram poklapanja može se specificirati težina 1 za 1-gram i 0 za 2, 3, 4 (1, 0, 0,0).
- **Kumulativni n-gram**: Računanje individualnih n-gram skorova u svim redosledima od 1 do n i postavljanje težina računanjem geometrijske sredine. Metoda *corpus_bleu()* računa komulativne 4-gram BLEU skorove – BLEU-4. Težine za BLEU-4 su ¼ za 1-gram, 2-gram, 3-gram, 4-gram skorove. Poželjno je izraćunati kumulativne BLEU-1 do BLEU-4 skorove prilikom evaluacije generisanog teksta.

Rezultati

Sledeća tabela prikazuje dobijene BLEU skorove za svaki testirani model posebno. Može se uočiti da svim modeli daju relativno slične vrednosti ove metrike, pri cemu se izdvajaju modeli koji ne koriste GloVe u embeding sloju u odnosu na one koji koriste.

VGG-16	*BLEU-1* 0.585 *BLEU-2* 0.358 *BLEU-3* 0.265 *BLEU-4* 0.142
InceptionV3	*BLEU-1* 0.593 *BLEU-2* 0.355 *BLEU-3* 0.255 *BLEU-4* 0.131
VGG-16 + GloVe embedding	*BLEU-1* 0.567 *BLEU-2* 0.341 *BLEU-3* 0.250 *BLEU-4* 0.131
InceptionV3 + GloVe embedding	*BLEU-1* 0.576 *BLEU-2* 0.346 *BLEU-3* 0.253 *BLEU-4* 0.133

Sledeća tabela pokazuje učinkovitost datih modela pri predikciji opisa nad novim slikama. Slike su izabrane tako da se može demonstrirati gde su modeli uspešni, a pri kojim tipovima slika greše. Može se primetiti da svi modeli uspešno prepoznaju pse i decu, što je posledica velike količine slike u trening skupu podataka koji prikazuju ove objekte. Takođe, može se primetiti da većina modela pravi asocijaciju između pasa i trave/lopte, čak i kada je ta asocijacija pogrešna, što se takođe može prepisati kontekstu slika korišćenog skupa podataka. Interesantno je da, svi modeli apsolutno greše pri opisu četvrte slike, dvoje ljudi koji na travi čitaju knjigu.

VGG-16	three girls are sitting in front of grass
IncV3	two children are sitting on the camera
VGG-16 + GloVe	two children sit on the street
IncV3 + GloVe	two girls are sitting on the grass

VGG-16	white dog is running in the air
IncV3	two dogs are playing in the grass
VGG-16 + GloVe	white dog is running through the grass
IncV3 + GloVe	two dogs are playing with ball in the snow

VGG-16	two dogs are running through the grass
IncV3	two dogs are playing in the grass
VGG-16 + GloVe	two dogs run through the grass
IncV3 + GloVe	two dogs are playing with ball in the grass

VGG-16	child is jumping on the grass
IncV3	two dogs are playing in the grass with his arms
VGG-16 + GloVe	little boy in yellow shirt is sitting on the grass
IncV3 + GloVe	two young boys are playing with large bag

VGG-16	two children are playing in the field
IncV3	young boy in pink shirt is running on the grass
VGG-16 + GloVe	two girls are playing in field
IncV3 + GloVe	young boy in pink shirt is running through the grass

Poređenjem BLEU skorova bazičnih modela možemo videti da najbolji skor daje model koji koristi *InceptionV3* konvolucionu mrežu, dok je na drugom mestu model koji koristi *VGG-16* mrežu. Takođe poređenjem rezultata dobijenim generisanjem opisa pojedinaćnih slika možemo zaključiti da bazični model koji koristi *VGG-16* mrežu generiše uverljivije opise slika. Što se tiče modela koji koriste *GloVe*, model koji koristi *InceptionV3* daje najbolji BLEU skor, bolji od modela

koji koristi *VGG-16*, ali poređenjem opisa fotografija možemo videti da i ovog puta *InceptionV3* daje gore opise. Ovo je, kao i u prethodnom bazičnom modelu, najverovatnije, posledica manje preciznosti *InceptionV3* modela. Poređenjem rezultata *VGG-16* bazičnog modela i *VGG-16 + GloVe* modela može se zaključiti da bazični model daje za nijasnu bolje i preciznije opise, što je u skladu sa njegovim boljim BLEU skorom. Na kraju, može se zaključiti da bazični *VGG-16* model generiše najbolje opise fotografija.

Zaključak

Ovaj rad bavio se problemom generisanja opisa slika korišćenjem tehnika dubokog učenja. Predstavljene su potencijalne arhitekture mreža koje se mogu koristiti u ove svrhe, naime modeli bazirani na enkoder-dekoder mrežama: *injected* i *merged* varijante modela. Ovaj rad, i implementacija data uz njega, ograničili su se na *merged* tip arhitekture. U okviru date arhitekture, rad je vršio eksperimente sa korišćenjem različitih tipova konvolucionih neuronskih mreža u enkoder delu, kao i razlika u implementaciji *embedding* sloja. Kao metrika evaluacije korišćen je BLEU skor, kao i subjektivna procena valjanosti modela na osnovu opisa generisanih na prethodno ne viđenim fotografijama.

Na osnovu rezultata prikazanih u ovom radu može se zaključiti da dati modeli daju međusobno relativno slične rezultate, pri čemu se kao najuspešniji pokazuje model koji kao enkoder koristi VGG-16 mrežu i ne koristi GloVe embeding. Zajedničko za sve modele je da funkcionišu bolje na slikama koje „liče“ na one iz trening skupa podataka, kao i da imaju tendenciju za povezivanje objekata koji se učestalo pojavljuju zajedno (iako ta asocijacija u datoj slici ne postoji). Ovakvo ponašanje je do negde očekivano, s obzirom na prirodu i veličinu korišćenog skupa podataka za treniranje modela. Što se tiče same lingvističke smislenosti generisanih opisa, primećuje se da svi modeli generisu smislene opise.

Finalno, može se zaključiti, da se primenom modela datih u okviru ovog rada, ne mogu pouzdano generisati opisi velikog raspona fotografija, ali da u određenim „klasama“ fotografija mogu davati solidne rezultate i pružati asistenciju pri kreiranju opisa slika.

Literatura

[1]: M. Tanti at al., “Where to put the Image in an Image Caption Generator,” in *Natural Language Engineering*, vol. 24, 2017.

[2]: M. Tanti at al., “What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?,” in *Proceedings of the 10 th International Conference on Neural Language Generation*, 2017.

[3]: O. Vanyals, “Show and Tell: A Neural Image Caption Generator,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3156-3164, 2015.

[4]: J. Brownlee, “Caption Generation with Inject and Merge Encoder-Decoder Models,” 2017. [Online]: https://machinelearningmastery.com/caption-generation-inject-merge-architectures-encoder-decoder-model/.

[5]: J. Brownlee, “A Gentle Introduction on Calculating the BLEU Score for Text in Python,” 2017. [Online]: https://machinelearningmastery.com/calculate-bleu-score-for-text-python/.