Automatsko prevođenje

Vladimir Vuksanović

Matematički fakultet, Beograd

Sadržaj

1 Uvod

Automatsko prevođenje podrazumeva prevođenje teksta sa jednog prirodnog jezika na drugi korišćenjem kompjutera. Preciznije, potrebno je za ulaznu rečenicu proizvoljne dužine na izvornom jeziku generisati izlaznu rečenicu, ne nužno iste dužine, na ciljnom jeziku koja ima isto značenje kao ulazna.

Klasičan način rešavanja ovog problema je statistički zasnovan, SMT (Statistical Machine Translation), pristup koji je bio dominantan decenijama i davao je prilično dobre rezultate. Međutim, u poslednjih nekoliko godina fokus je prešao na modele zasnovane na neuronskim mrežama, prvo kao dopuna SMT modelima pa onda i kao odvojenu paradigmu koja je eventualno uspela da nadmaši klasične pristupe.

U ovom radu će biti predstavljena pojednostavljena varijanta modela za prevođenje zasnovanog na neuronskim mrežama opisanog u [8] koji jedan od prvih radova u kome je predložena enkoder-dekoder arhitektura koja se i danas najčešće primenjuje.

2 Rešenje

Za rešavanje ovog problema ćemo koristiti NMT (Neural Machine Translation) pristup koji koristi neuronsku mrežu da predvidi sekvencu reči u prevodu. Kako su ulaz i izlaz sekvence koje mogu biti proizvoljnih dužina, prevođenje pripada grupi sequence-to-sequence problema. To onemogućava korišćenje potpuno povezane mreže jer ona zahteva fiksan broj ulaza i izlaza. Umesto toga, moderan pristup koristi enkoder-dekoder arhitekturu opisanu u [8] i [4] u kombinaciji sa rekurentnim neuronskim mrežama (LSTM ili GRU). Ova arhitektura u praksi daje veoma dobre rezultate zbog čega je i koristi jedan od najpoznatijih servisa ovog tipa, Google Translate [9].

Osnovna ideja je da se model podeli na dva dela. Prvi deo, enkoder, prima ulaz reč po reč (ili slovo po slovo), svaku od njih prosleđuje LSTM sloju i na osnovu toga generiše reprezentaciju *fiksne* dužine koja sadrži iste informacije kao polazna rečenica samo mapirana u novi prostor. Ta reprezentacija se zatim prosleđuje drugom delu, dekoderu, kao polazna tačka za generisanje prevoda na ciljni jezik analognim postupkom, reč po reč. U svakom koraku ulaz u dekoder čine trenutno stanje LSTM sloja i poslednja generisana reč. Na početku ta reč je poseban token za početak rečenice, a reči se generisu sve dok se ne dođe do tokena za kraj prevoda ili do zadate maksimalne dužine.

Originalni rad je predložio da se obrtanjem redosleda reči u ulaznim rečenicama bolje očuvaju veze između reči, ali kako su rečenice na kojima je ovaj model treniran kratke, ovaj doprinos se ne primećuje i neće biti korišćen.

3 Implementacija

Model je implementiran u programskom jeziku Python uz korišćenje keras biblioteke. Implementacija prati osnovne korake sa zvaničnog sequence-to-sequence primera sa keras veb stranice [1] pri čemu se predikcija vrši na nivou reči umesto pojedinačnih slova. U nastavku će biti opisan postupak pretprocesiranja i građenja modela za treniranje i predviđanje zajedno sa parametrima koji su korišćeni za treniranje.

3.1 Pretprocesiranje

Podaci za trening se sastoje od rečenica na izvornom jeziku i njihovih prevoda na ciljni jezik. Pre njihove obrade, izbačeni su svi parovi koji imaju isti ulaz a drugačiji prevod. Ovo je odrađeno zato što skup podataka sadrži veliki broj duplikata koji međusobno ometaju jedni druge.

Pošto direktna upotreba rečenica u tekstualnom obliku nije pogodna, potrebno je pretvoriti ih u reprezentaciju koja bolje radi sa neuronskom mrežama. Prvi korak pretprocesiranja je eliminisanje specijalnih simbola iz rečenica jer nam oni nisu korisni prilikom prevođenja. Zatim, kako bi rečenice mogle da se upotrebe kao ulaz neuronske mreze potrebno je predstaviti ih kao niz brojeva. To se radi jednostavnim pridruživanjem broja svakoj reči iz trening skupa ili nekog predefinisanog rečnika u kom slučaju se dodaje i token za nepoznate (out-of-vocabulary) reči. Ovaj proces se naziva tokenzacija i radi se odvojeno za ulaz i izlaz. Za velike trening skupove bi bilo bolje ograničiti skup reči na rečnik, ali ovde je korišćen pristup sa svim rečima iz trening skupa.

Originalni papir predlaže dodatno da se za ulazne rečenice obrne redosled tokena u sekvenci zato što se tako bolje očuvaju veze između reči u dužim rečenicama. Zbog kratkih rečenica u trening skupu, ovu tehniku nećemo primeniti, ali je ostavljena kao opcija u kodu.

Dalje, potrebno je napraviti dva tipa izlaznih rečenica, jedan tip predstavlja ulaz u dekoder a drugi predstavlja izlaz. Prva reč koju dekoder treba da dobije je početni (start of sequence) token i on se dodaje na početak svake rečenice koja ulazi u dekoder. Slično, na rečenice koje predstavljaju izlaz iz dekodera potrebno je dodati završni (end of sequence) token.

Sve sekvence iz respektivnih skupova se potom dopunjavaju "praznim"tokenima tako da budu iste dužine. Pri tome, ulazne sekvence se dopunjuju sa leve strane jer se konačan rezultat dobija pri izlazu poslednjeg tokena, a izlazne sekvence se dopunjuju sa desne strane jer njihova obrada počinje prvim tokenom. Ovim smo dobili skup ulazno-izlaznih sekvenci koje mogu da se koriste za treniranje neuronske mreže.

3.2 Enkoder

Enkoder je zadužen da na osnovu dobijene sekvence generiše reprezentaciju fiksne dužine koja bi trebalo da sadrži sve informacije potrebne da se konstruiše prevod na ciljnom jeziku.

Ulaz u enkoder je tensor dimenzije $n \times m$ gde je n broj rečenica, a m dužina ulazne sekvence. On je vezan za embedding sloj koji svakoj reci dodeljuje vektor fiksne dužine tako da udaljenost između vektora zavisi od značenja reči. Umesto da model uči te reprezentacije za vreme treninga, koristimo GloVe model [6] treniran na 5 milijardi reči sa vikipedije gde je svaka reč prestavljena vektorom dužine 100. Embedding sloj je zatim vezan na LSTM sloj dimenzije 512 koji se inicijalizuje nasumičnim vrednostima iz uniformne raspodele na intervalu $[-0.08, 0.08]$. Kao izlaz iz ovog sloja se dobija njegovo interno stanje na kraju koje prestavlja novu reprezentaciju ulazne rečenice.

3.3 Dekoder

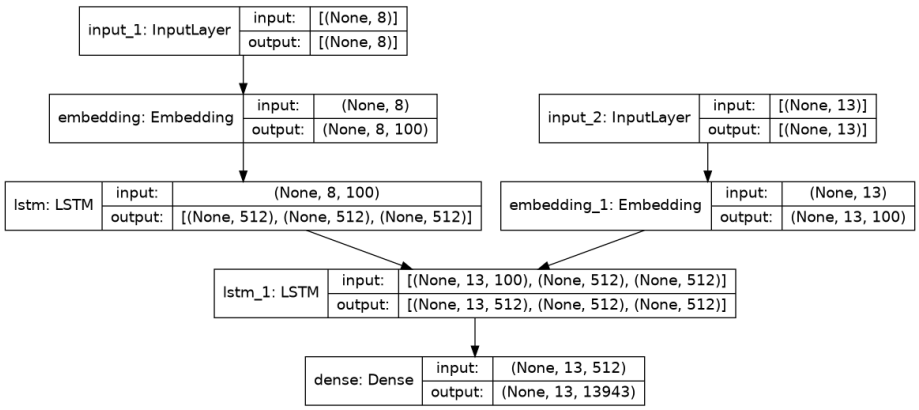
Dekoder koristi svoje unutrašnje stanje i prethodno generisanu reč kako bi odredio sledeću reč u rečenici.

Po strukturi, dekoder je veoma sličan enkoderu. Ulazni sloj sada ima drugu dimenziju jednaku dužini izlazne sekvence dok embedding i LSTM slojevi ostaju nepromenjeni osim sto težine za embedding sloj neće biti unapred zadate nego će se učiti u toku treniranja. Na kraju se nadovezuje na potpuno povezani sloj sa softmax aktivacijom koji određuje verovatnoću za svaku reč na ciljnom jeziku.

Inicijalno se kao unutrašnje stanje koristi izlaz iz enkodera, a kao prethodno generisanu rec uzimamo početni (<sos>) token koji je uveden prilikom pretprocesiranja. Dekodiranje se prekida ili kada se dobije završni (<eos>) token ili dok se ne dostigne zadata maksimalna dužina rečenice.

3.4 Model

Kombinovanjem prethodno opisanih enkodera i dekodera tako da dekoder inicijalno dobije stanja iz enkodera, dobija se konačni model prikazan na slici 1.



Slika 1: Model za treniranje

Optimizator koji koristimo je *rmsprop*, funkcija gubitka je kategorijska unakrsna entropija, a metrika koju pratimo je tačnost.

3.5 Trening

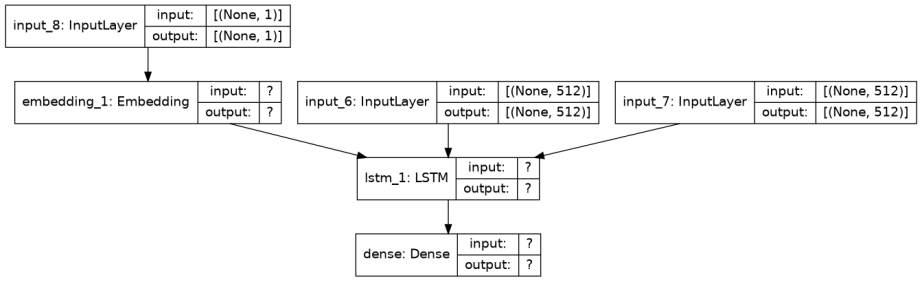
Opisani model je treniran na 50.000 parova rečenica na engleskom i francuskom jeziku iz skupa preuzetog sa [2]. Trening je trajao 5 epoha sa veličinom grupe od 128 instanci i 10% skupa je korišćeno za validaciju. Pošto ceo skup obrađenih podataka ne moze da stane u radnu memoriju za veće brojeve rečenica, korišćen je generator da se grupe rečenica konstruišu tek kada su potrebne.

Za treniranje je korišćena Google Colaboratory platforma i trajalo je 2 sata.

3.6 Generisanje prevoda

Prethodni model radi samo ako već na samom početku imamo celu prevedenu rečenicu što je dobro za vreme treniranja modela ali je neupotrebljivo

za potrebe prevođenja proizvoljnih novih rečenica. Iz tog razloga pravimo novi model koji koristi već trenirane delove iz prethodnog modela ali sa modifikovanim dekoderom. Ulaz u novi dekoder je samo jedna reč koja se inicijalno postavlja da bude početni token. Sada je moguće iterativno pozivati dekoder sve dok ne dođe do zavrsnog tokena ili nekog unapred određenog limita na dužinu rečenice. Novi model je prikazan na slici 2.



Slika 2: Model za predviđanje

3.7 Beam search algoritam

Umesto da se prevod traži gramzivo, uvek gledajući jedan mogući prevod, možemo da dobijemo veliko poboljšanje koristeći *beam search* algoritam koji će heuristički da proveri veći broj prevoda i izabere najbolji među njima.

Prvo se zadaje parametar k koji predstavlja broj rečenica koje će se razmatrati u svakom trenutku. Na početku se bira k reči koje mreža predviđa kao najverovatnije na osnovu vrednosti potpuno povezanog sloja. Kada se izabere prvih k kandidata, svaki od njih se potom dopunjuje svim mogućim rečima i bira se k najverovatnijih sevenci među njima. Ovaj postupak se ponavlja do kraja svake od sekvenci i onda se za konačan prevod uzima najbolja od njih. Specijalno, za vrednost $k = 1$ postupak se svodi na gramzivu pretragu.

Formalno, algoritam maksimizira vrednost funkcije

$$\arg\max_y \prod_{t=1}^{T_y} P(y^{<t>}|X, y^{<1>}, y^{<2>}, ..., y^{<t-1>})$$

Kako su sve verovatnoće između 0 i 1, prethodni proizvod brzo postaje veoma mali broj. Zbog toga se umesto običnog proizvoda koristi suma logaritama.

$$\arg\max_y \sum_{t=1}^{T_y} \log P(y^{<t>}|X, y^{<1>}, y^{<2>}, ..., y^{<t-1>})$$

Sada kako je suma logaritama strogo rastuća funkcija, prethodna formula favorizuje što duže rečenice. Da bi kompenzovali za ovo, normalizovaćemo formulu tako što je podelimo sa brojem reči u prevodu. Ovo je konačna formula koju koristimo.

$$\arg\max_y \frac{1}{T_y} \sum_{t=1}^{Ty} \log P(y^{<t>}|X, y^{<1>}, y^{<2>}, ..., y^{<t-1>})$$

Od izbora parametra k zavisi koliko potencijalnih prevoda će biti razmatrano i time celokupni kvalitet prevoda. Što je parametar veći, više izbora se razmatra ali se vreme za dobijanje prevoda ekponencijalno povećava. U implementaciji je korišćena vrednost $k = 3$. Više detalja se može naći u [5] i [7].

4 Rezultati

4.1 Primeri prevoda

U nastavku su prikazani primeri prevoda rečenica iz trening skupa gramzivim i beam search algoritmom kao i njihov tačan prevod.

Recenica	What's that dog doing?
Tacan prevod	Que fait ce chien ?
Model (greedy)	de quoi chien est il
Model (beam)	qu'est ce que ce chien

Recenica	That boy is smart.
Tacan prevod	Ce garçon est intelligent.
Model (greedy)	ce garçon est intelligent
Model (beam)	ce garçon est intelligent

Recenica	Tom tasted the cake.
Tacan prevod	Tom a goûté le gâteau.
Model (greedy)	tom a mangé le gâteau
Model (beam)	tom a mangé le gâteau

Recenica	It's like riding a bike.
Tacan prevod	C'est comme le vélo, ça ne s'oublie pas.
Model (greedy)	c'est comme comme comme un vélo
Model (beam)	c'est comme comme un vélo

Recenica	You can't bury the truth.
Tacan prevod	Vous ne pouvez pas enterrer la vérité.
Model (greedy)	vous ne pouvez pas les autres
Model (beam)	vous ne pouvez pas la vérité

Recenica	I know that I'm rambling.
Tacan prevod	Je sais que je suis incohérent.
Model (greedy)	je sais que c'est un peu
Model (beam)	je sais que je suis en colère

Recenica	I neither drink nor smoke.
Tacan prevod	Je ne bois ni ne fume.
Model (greedy)	je ne bois plus de ne bois
Model (beam)	je ne bois rien

Može se primetiti da je model počeo da povezuje reči na engleskom sa rečima na francuskom, pa čak nekada i zamenjuje reč nekom sličnom, ali za reči koje

očekujemo da se retko javljaju u skupu za treniranje često daje neku drugu nasumičnu reč ili potpuno izgubi smisao.

U narednim tabelama se nalaze primeri prevoda rečenica koje nisu iz trening skupa poređane po kompleksnosti od jedne reči do komplikovanijih rečenica.

Recenica	You
Tacan prevod	Tu
Model (greedy)	tu as
Model (beam)	tu es

Recenica	It's raining.
Tacan prevod	Il pleut.
Model (greedy)	il pleut
Model (beam)	il pleut

Recenica	What time is it?
Tacan prevod	Quelle heure est il ?
Model (greedy)	à quoi ça est le temps
Model (beam)	à quelle heure est il

Recenica	I am going to run home tomorrow.
Tacan prevod	Je vais courir à la maison demain.
Model (greedy)	je vais aller à la maison
Model (beam)	je vais aller à la maison

4.2 BLEU ocena

BLEU (The Bilingual Evaluation Understudy) ocena je mera za ocenu kvaliteta mašinskog prevoda tako što ga poredi sa ljudski generisanim prevodima. Ocena ima vrednost iz opsega 0 do 1 gde 0 predstavlja potpuno pogrešan, a 1 savršen prevod.

U nastavku su tabele sa nasumično izabranim recenicama iz trening skupa i njihove ocene.

Recenica	You're polite.	
Tacan prevod	tu es poli	
Model (greedy)	tu es intelligent	$1.13e^{-154}$
Model (beam)	tu es intelligent	$1.13e^{-154}$

Recenica	That sounds exhausting.	
Tacan prevod	ça a l'air épuisant	
Model (greedy)	ça a l'air bizarre	$8.63e^{-78}$
Model (beam)	ça a l'air bizarre	$8.63e^{-78}$

Recenica	I'm as tall as my father.	
Tacan prevod	je suis aussi grand que mon père	
Model (greedy)	je suis aussi grand que moi	0.64
Model (beam)	je suis aussi grand que moi	0.64

Recenica	Tom wasn't sure.	
Tacan prevod	tom n'était pas sûr	
Model (greedy)	tom n'est pas sûr	$1.05e^{-154}$
Model (beam)	tom n'est pas sûr	$1.05e^{-154}$

Recenica	I didn't sleep.	
Tacan prevod	je n'ai pas dormi	
Model (greedy)	je n'ai pas dormi	1.0
Model (beam)	je n'ai pas dormi	1.0

Vidimo da u velikom broju slučajeva ocena prevoda je blizu 0 što je i očekivano jer se ocenjivanje vrši na osnovu preklapanja n-grama a model osim što radi sa kratkim rečenicama obično pogreši bar jednu reč. Prve dve rečenice imaju pogrešno prevedenu po jednu reč, u trećoj nedostaje jedna, u četvrtoj je upotrebljeno pogresno vreme glagola dok je poslednja potpuno tačna. Donekle iznenađujuće, sve rečenice su gramatički ispravne.

5 Zaključak

Na osnovu rezultata iz prethodnog poglavlja možemo zaključiti da čak i sa skupom od nekoliko desetina hiljada rečenica, što je malo za ovako složen problem, model razume veze između reči na izvornom i ciljnom jeziku koje se često ponavljaju. Problem nastaje kada treba da prevede reči koje se retko nalaze na trening skupu što moze biti popravljeno korišćenjem skupa sa manjim brojem reči ali većim brojem rečenica. Jedna preporuka za skup podataka ukoliko je cilj napraviti ozbiljniji prevodilac je [3] koji se koristi kao standardni test za evaluaciju složenih modela. Takođe, kao što je preporučeno u [8] dodavanjem više LSTM slojeva znatno se povećava mogućnost razumevanja modela po cenu vremena treniranja. Taj rad je koristio po 4 LSTM sloja za enkoder i dekoder, dok je [9] koristio čak 8.

Literatura

[1] URL: https://keras.io/examples/nlp/lstm_seq2seq/.

[2] URL: http://www.manythings.org/anki/.

[3] URL: https://www.tensorflow.org/datasets/catalog/wmt14_translate.

[4] Kyunghyun Cho **andothers**. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: 1406.1078 [cs.CL].

[5] Markus Freitag **and** Yaser Al-Onaizan. „Beam Search Strategies for Neural Machine Translation". **in**: *Proceedings of the First Workshop on Neural Machine Translation* (2017). DOI: 10.18653/v1/w17-3207. URL: http://dx.doi.org/10.18653/v1/W17-3207.

[6] Jeffrey Pennington, Richard Socher **and** Christopher D. Manning. „GloVe: Global Vectors for Word Representation". **in**: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, **pages** 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

[7] Stuart Russell **and** Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd. USA: Prentice Hall Press, 2009. ISBN: 0136042597.

[8] Ilya Sutskever, Oriol Vinyals **and** Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: 1409.3215 [cs.CL].

[9] Yonghui Wu **andothers**. *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. 2016. arXiv: 1609.08144 [cs.CL].