Nenadgledano Učenje Reprezentacije Govora Kroz Predikciju Zvuka

Igor Petrović
Fakultet tehničkih nauka
Univerzitet u Novom Sadu
Trg Dositeja Obradovića 6
21000 Novi Sad
igor.be.petrovic@gmail.com

***Apstrakt*—Jedan od problema koji se često sreće u obradi govora je mala količina labeliranih podataka. S obzirom da se na internetu nalazi velika količina nelabeliranih snimaka govora, bilo bi korisno nenadgledano naučiti reprezentaciju koristeći ovu veliku količinu podataka, i samo *finetune*-ovati je za konkretan problem na relativno malom skupu za obuku. Insipirisan prvobitnim GPT [1] radom, koji je ovaj pristup primenio autoregresivnom predikcijom tokena teksta, u ovom radu se koristi sličan pristup za obuku generativnog modela govora, i njegovog *fine-tune*-ovanja na zadatak transkripcije srpskog jezika. Tokenizacija zvuka se vrši primenom *Vector Quantized Variational Autoencoder* [2] (VQ-VAE), koji se zatim predviđa pomoću generativnog transformer [3] modela. Konačno, reprezentacija poslednjeg sloja generatora se koristi kao tenzor obeležja prilikom obuke konvolucionog sloja koji vrši predikciju karaktera, optimizovanu kroz *Connectionist Temporal Classification* (CTC) *loss* funkciju. Konačni rezultati postižu 65% Levenshtein [4] ratio na test skupu, kada je konvolucioni sloj obučavan na 15 minuta labeliranih podataka.**

***Ključne Reči*—nenadgledano učenje, genertivno modelovanje, VQ-VAE, Transformer, transkripcija govora, srpski jezik**

I. O Predikciji Zvuka

Problem koji se javlja kod predikcije zvuka je to što je, za razliku od teksta, zvuk kontinualan, i različite kontinuacije se ne mogu trivijalno reprezentovati kao distribucija verovatnoće. Kako bi se ovaj problem prevazišao, inspirisan *Vector Quantized Variational Autoencoder* (VQ-VAE) radom, koji se bavi problemom tokenizacije slika, zvuk se može pretvoriti u mel spektrogram, koji bi se zatim, nalik na sliku, tokenizovao VQ-VAE modelom. Ovako tokenizovan zvuk govora se dalje može ispraviti u jednodimenzioni niz tokena, koji je tada prilagođen obliku za generativnu obuku pomoću transformer modela.

II. Podaci

Za ovaj rad, korišteni su podaci preuzeti sa interneta, koji se sastoje od 29 epizoda srpskih podkasta, što ukupno čini oko 70 sati zvuka. Ovi podaci su podeljeni na trening i test skupove, gde prvih 90% svakog audio zapisa čini trening skup, a poslednjih 10% test skup. Test skup nije korišten tokom obuke nijednog od modela kako bi se obezbedila nepristrasna evaluacija performansi.

A. Augmentacija Podataka

Trening skup je zatim augmentovan kako bi se povećala raznovrsnost podataka i poboljšala robusnost modela. Augmentacija je izvršena na sledeći način:

- **Randomizacija brzine:** Brzina zvuka je nasumično varirana u opsegu (0.9, 1.1).
- ***Pitch*-shiftovanje:** *Pitch* zvuka je promenjen korištenjem praat biblioteke u opsegu (0.5, 1.5).

Svaki snimak je augmentovan ovim tehnikama, pri čemu je generisano 10 novih verzija svakog snimka. Na taj način, trening skup je uvećan desetostruko, što je značajno poboljšalo kapacitet modela za učenje različitih varijacija govora.

III. Arhitektura Modela

Arhitektura modela sastoji se iz dva osnovna dela: VQ-VAE za tokenizaciju zvuka, i autoregresivnog *decoder-only* transformera za predikciju zvuka.

A. Tokenizacija Zvuka

Prvi korak je pretvaranje zvuka govora u mel spektrogram, dvodimenzioni prikaz zvuka gde jedna osa predstavlja vreme, a druga frekvenciju. Ovaj spektrogram postaje ulaz u VQ-VAE model, koji je treniran da kvantizuje spektrogram u diskretne tokene. Konkretno, svaka sekunda zvuka se predstavlja kao tenzor veličine 8×32 tokena sa vokabularom od 8192. Ovaj tenzor se zatim ispravlja u jednodimenzioni niz od 256 tokena.

B. Treniranje VQ-VAE Modela

Sirovi zvuk se najpre pretvara u mel spektrogram. VQ-VAE model se zatim trenira da kvantizuje mel spektrogram u diskretne tokene. Trening se vrši minimizacijom *loss* funkcije koja se sastoji iz rekonstrukcionog *loss*-a i VQ *loss*-a. Rekonstrukcioni *loss* osigurava da kvantizovani spektrogram može biti rekonstruisan nazad u originalni spektrogram sa minimalnim gubicima, dok VQ *loss* obučava embeding vektore rečnika. Prilikom rada na arhitekturi ovog modela, implementirani su hiperparametri *horizontal downscale* (`hdown`) i *vertical downscale* (`vdown`). Svaki inkrement ovih parametara prepolovljava rezoluciju po toj dimenziji. Sa većim brojem se postiže i veća kompresija (manji broj tokena), ali i lošiji kvalitet prilikom rekonstrukcije zvuka. Treniranjem više modela sa različitim kombinacijama ovih parametara, i

osiguravanjem da se zvuk dobro razaznaje, konačne vrednosti ovih parametara postavljenje su na `hdown=1` i `vdown=4`, tako da je spektrogram originalne rezolucije 128 (mela) $\times$ 64 za jednu sekundu zvuka postao 8×32 tokena. Veličina embeding vektora tokom obuke VQ-VAE-a je postavljena na 256, broj filtera konvolucionih slojeva na 256, i sadrži 2 rezidualna konvoluciona sloja. Model je obučavan sa MSE loss funkcijom, veličinom beča 32 i stopom učenja 3e-4 sa Adam optimizatorom.

C. Treniranje Generativnog Transformer Modela

Nakon što je VQ-VAE model istreniran, koristi se da tokenizuje sopstveni trening skup, ispravi ga u niz, i izveze u zaseban tekstualni fajl. Generisani nizovi tokena iz ovog fajla se zatim koriste za treniranje *decoder-only* transformer modela. Ovaj model je treniran da predviđa sledeći token na osnovu prethodnih tokena u nizu. Obučava se minimizacijom *loss*-a unakrsne entropije pravog sledećeg tokena. Arhitektura, kao i režim obučavanja ovog modela preuzeti su iz rada Whisper modela [5], a treniran je Chinchilla [6] zakonima skaliranja. U okviru rada, istrenirana su dva modela različitih veličina (15M i 100M parametara) sa dužinom konteksta 2048 (što odgovara 8 sekundi zvuka). Modeli su obučavani AdamW optimizatorom sa `weight decay=0.1`, dok je stopa učenja postepeno podizana od faktora 0.1 inicijalne vrednosti, do finalne u okviru 2000 *warmup* koraka (radi stabilnije obuke), a zatim je smanjivana kosinusnim *scheduler*-om do faktora 0.1 svoje maksimalne vrednosti tokom ostatka treninga po Chinchilla receptu skaliranja [6]. Model od 100M parametara je treniran 7 dana na RTX 3090. Ostatak hiperparametara se nalazi u apendiksu.

IV. Finetuning za Transkripciju

Nakon što je generativno transformer model istreniran, može se *finetune*-ovati za zadatak transkripcije. U ovom koraku, izlaz transformer modela (bez poslednjeg linearnog sloja) se koristi kao reprezentacija zvuka. Kako je dužina konteksta generativnog tranformera 2048 (8×256), svaki osmi token predstavlja poslednji token te tačke u vremenu. Zbog toga, umesto da se koristi čitava reprezentacija generatora, mogu se koristiti samo osmi embeding vektori - što smanjuje dužinu sekvence na 256.

A. Arhitektura Transkripcionih Modela

Inicijalna ideja prilikom izrade ovog projekta je bila koristiti transformer dekoder za transkripciju, tako što bi preko *cross-attention*-a pristupao izlaznim embeding vektorima generatora, i tako vršio *sequence-to-sequnce* transkripciju. Problem kod ovog pristupa bio je to što transformeri zahtevaju veliku količinu podataka za obuku, i transkripcioni dekoder se vrlo brzo natprilagodio trening podacima - memorišući trening sekvence. Čak i kada je `weight decay` parametar postavljan na jako visoke vrednosti, dekoder je prestao da radi dobro na trening primerima, bez značajnog pomaka u rezultatima na validacionom skupu.

Kako bi se ovaj problem prevazišao, umesto transformer dekodera, treniran je jednodimenzioni konvolucioni sloj sa veličinom filtera 3 i *padding*-om 1.

B. Priprema Finetuning Podataka

Iz (već augmentovanog) trening skupa, na kojem su obučavani prethodni modeli, ekstrahuje se 110 osmosekundnih segmenata, koji su zatim ručno transkribovani. Ovi audio-tekst parovi čine osnovu *finetuning* skupa. Da bi se veštački povećala raznovrsnost ovih podataka, oni su dodatno augmentovani na sledeći način:

- **Randomizacija brzine:** Brzina zvuka je nasumično varirana u opsegu (1, 1.25).
- ***Pitch*-shiftovanje:** *Pitch* zvuka je promenjen u opsegu (0.5, 1.5).
- **Promena glasnoće:** Glasnoća zvuka je promenjena u opsegu (-20dB, +5dB).

Ova augmentacija je izvršena 10 puta za svaki audio segment, kreirajući ukupno 1100 osmosekundnih segmenata, ili oko 2.5 sata audio sadržaja.

C. Finetuning Proces

Finetuning proces uključuje treniranje linearnog i konvolucionog modela na ovom malom skupu labeliranih audio-tekst parova. Model uči da mapira zvučne reprezentacije na odgovarajući tekst, optimizovan CTC *loss* funkcijom sa AdamW optimizatorom i stopom učenja 1e-4.

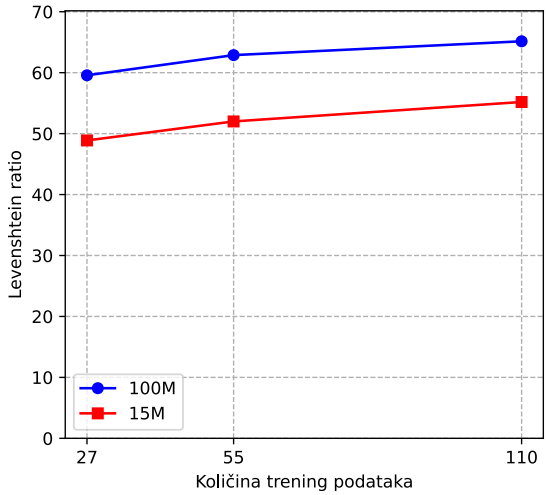
V. Evaluacija

Da bi evaluacija bila moguća, prvo je potrebno kreirati validacioni i test skup, kao i izabrati odgovarajuću metriku. Korišćenjem inicijalno izdvojenog trening skupa (10% ukupnog skupa audio zapisa), koji pritom nije augmentovan, izdvaja se (nakon čišćenja) 28 osmosekundnih segmenata za validacioni skup i 26 segmenata za test skup. Zatim se, ručnom transkripcijom, dolazi do labela za ove segmente.

Pošto CTC *loss* mapira slova na vremenske trenutke i nema jasan koncept razmaka, dobijeni rezultati se konkateniraju u jedan string bez razmaka, i uklanjaju se svi uzastopno ponovljeni karakteri. Kako bi poređenje bilo moguće, ista ova transformacija se obavlja i na tačnim labelama. Nakon toga, moguće je uporediti sličnost ovih stringova merenjem broja transformacija potrebnih da se od jednog dođe do drugog, i deljenjem ovog broja sa dužinom stringa. Dakle, kao metrika se može koristiti *Levenshtein*-ov *ratio*. Tokom treninga modela korišteno je rano zaustavljanje, i na test skupu su evaluirani modeli koji su trenirani sa brojem epoha koji daje najbolje rezultate na validacionom skupu.

VI. Rezultati

Oba modela (15M i 100M) su evaluirani na ovom testnom skupu, i izmerene su im performanse sa različitim procentom trening podataka. Testirani su sa 28, 55 i 110 labeliranih trening primera, ali je u u slučaju manjeg broja primieraka od 110, generisano više augmentacionih primera, tako da je konačan broj trening primeraka za svaki od ovih slučajeva bio oko 1100.



Sa grafika se može videti da 100M model ima značajno veću tačnost od 15M modela, čak i kada se trenira na 4 puta manje labeliranih podataka. Takođe se može videti da oba modela postaju predvidivo bolja sa više labeliranih trening podataka. Ovim se uočavaju dva puta za poboljšanje performansi modela. Jedan je povećanje broja trening podataka, dok je drugi skaliranje baznog modela i skupa za nenadgledanu obuku.

VII. Zaključak

Iako rezultati postignuti u ovom radu nisu dovoljno dobri za praktičnu upotrebu modela, sama činjenica da je moguće nenadgledano naučiti reprezentacije govora kroz generativnu predikciju omogućava primenu ovog pristupa u razne svrhe. U ranim fazama razvoja ovog rada, korišten je LSTM umesto transformera, i pokazalo se da je moguće razlikovati glasove dve osobe (prisutne u nelabeliranom korpusu) sa 100% tačnošću na malom testnom skupu kada se model *fine-tune*-uje na samo 10 sekundi glasa te osobe. Čak i sa 5 sekundi glasa, tačnost ostaje na visokih 95%. Što se konkretno transkripcije tiče, moguće je zamisliti velike modele, razvijene od strane firmi sa najviše resursa, koji sa 100B+ parametara, između ostalih modaliteta, mogu biti trenirani i na zvuku. Ovakvi modeli bi, pored tekstualnog modaliteta, mogli lako da rukuju i sa zvukom, i vrše prevođenje iz jednog oblika u drugi. To bi takođe otvorilo put ka velikoj količini dodatnih podataka za obučavanje modela.

Literatura

[1] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving Language Understanding by Generative Pre-Training, 2018. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[2] van de Oord, A., Oriol, V., Kavukcuoglu, K. Neural Discrete Representation Learning. *arXiv preprint arXiv:1711.00937*, 2017.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin I. Attention Is All You Need. *arXiv preprint arXiv:1706.03762*, 2017.

[4] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals of symbols", Dokl. Akad. Nauk SSSR, 163:4 (1965), 845–848.

[5] Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust Speech Recognition via Large-Scale Weak Supervision, 2022. https://cdn.openai.com/papers/whisper.pdf

[6] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training Compute-Optimal Large Language Models. *arXiv preprint arXiv:2203.15556*, 2022.

[7] Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Dhariwal, P., Luan, D., and Sutskever, I. Generative Pretraining from Pixels, 2020. https://cdn.openai.com/papers/Generative_Pretraining_from_Pixels_V2.pdf

[8] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-Shot Text-to-Image Generation. *arXiv preprint arXiv:2102.12092*, 2021.

[9] Yan, W., Zhang, Y., Abbeel, P., and Srinivas, A. VideoGPT: Video Generation using VQ-VAE and Transformers. *arXiv preprint arXiv:2104.10157*, 2021.

Apendiks

Parameter	15M	100M
nheads	6	12
dim	384	768
nlayers	4	10
ff size	1536	3072
learning rate	1.5e-3	5e-4
warmup steps	2000	2000
initial learning rate	1.5e-4	5e-5
weight_decay	0.1	0.1
optimizer	Adam	AdamW
beta1	0.9	0.9
beta2	0.98	0.98

TABLE I
HIPERPARAMETRI I KONFIGURACIJA ZA 15M I 100M MODELE