Context Bucketed Text Responses using Generative Adversarial Neural Network in Android Application with TensorFlow-Lite Framework

Prashant Kaushik, K. Vimal Kumar, Prantik Biswas
Department of CSE/IT
Jaypee Institute of Information technology,
Noida, Uttar Pradesh, India
Prashant.kaushik@mail.jiit.ac.in, vimalkumar.k@gmail.com, prantik.biswas@jiit.ac.in

2022 8th International Conference on Signal Processing and Communication (ICSC) | 978-1-6654-5430-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICSC56524.2022.10009634

Abstract— **One of the NLP's techniques is to train the Generative Adversarial Network (GAN) model by using Discriminator and generator which gets trained together and GAN generates outputs which are classified by discriminators as real or fake outputs. We on the other hands are using two more models with GAN named Contexture 1 and Contexture 2 to give a good result in the context of text input and using it for better user reactions for android based tensor-flow lite application which records user responses and stores them to our server for further training. In this paper we have wired generator, discriminator and two contextures such that we get an output from generator with a context and context score. Based on these context scores we align these responses to the users in android application which is running a TensorFlow-lite application for interacting with user. Our TC-GAN is primarily trained on input sentences from various data set like MNIST, NewsQA, SQuAD,CelebA etc. Also, its based-on user reaction a retraining is done on server, a new model is pushed to android application without updating the android application. This allow context-based user training and a fully featured framework which can be used in any application like bank chat bots and ecommerce chat bot where the text-based chat model is continuously evolving.**

Keywords—GAN, Tensor flow, context bucket

I. INTRODUCTION

Generative model for text generation is basically word embedding computed using diverse method, that is essential tool for natural language processing and information retrieval for different applications like text generation [16], image manipulation, dialogue representations etc. The text generation model normally deals with sequence to sequence [17] mode. For long text generation or word embedding system, there are many generative models that are successively in a progressive mode like seqGAN[10] uses Monte Carlo[1] search has between sample data to the remaining input and intermediate step before passing to MC search. But seqGAN has not achieved the desirable result for long text generation because uses the gradient based Reinforcement learning policy for reorganized the generative model.

Variation Auto-encoder (VAEs) [12] trains the latent representation, it is treated as latent variable during training for sentence representations. Variation auto-encoder is very effective when trained on text data. VAE mainly consist of training process and generation process for desirable result. In training process encoder use as an input with convolutional layer and decoder use as a output with deconvolutional layer, uses feed forward method for generated sentence by combining with recurrent layer. In this manner we use sentence encoder for make the effective model for natural language processing (NLP) task and we take the data from the various types of data set for training process to make the generated sentence context oriented or the desired output. In this model we are using the Gated recurrent unit (GRU) model for encoding the high-level data sentence and provide this information to decoder. GRU [17] is basically advanced model of long short-term memory (LSTM) model.

Presently there are many models that are working on question answer (Q/A) and reading comprehension [3] [10] data set. This model is for training sentence based labeled data. It is large task to sustain the question answer data set. Some of data set have only thousand pairs of Q/A for example Web questions, MCTest[19], WikiQA[14] , and TREC-QA[1]. Also, a large question answering data set they have more than thousand sets of data set like SQuAD[20], NewsQA[22] etc. are difficult task to store this data, because this issue makes task hamper the real world applications for particular specific question answering. Generative Domain- Adaptive networks (GDANS) [23] model mainly deals with semi-supervised question answer data set. In this architecture a discriminative classifier and predictor works in a alternating way between training and test distributions this process is known as domain, adaptation (DA). When tokens are given and fixed then the mappings between source and target domains are developed and then source domain works during training time and targets sentence in the test time for sentence representations. This model consists of two types of classifier first one is label predictor that predict the label in training time and test time and the second one is domain classifier that differentiate the training of target and source domain. Deep feature mapping minimizes the loss of label classifier and maximize the loss of domain classifier. During this process classifier act as a middle worker between source domain and target domain whether the token is either fully unlabeled (unsupervised domain annotation) or have few labeled tokens (semi-supervised domain adaptation). Previously most paper on domain adaption worked with feature representation deep learning with one training process is known as deep domain adaption. In GDANS model three major processes works together for embedding an appropriate network by using different layers and loss functions and this process is trained by back propagation algorithms.

978-1-6654-5430-8/22/$31.00 ©2022 IEEE

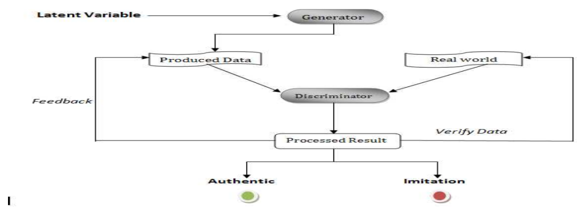

Figure 1: - Generative Adversarial Network (GAN) Model

II. OUR WORK

Our model is mainly divided into various sub sections which is discussed below for providing better information and details about key implementations done in tensor-flow. After transformation procedure of fully trained model is done, the main thing is to have keep model for faultlessness of on input and output dimensions. After that we can use this model into android application in which input is noise by user and generate output by two contextures in a manner of four context then deposit the user reaction on to our servers those are running in a firebase server framework. We have used large movie review dataset which is open source and then user collected responses are used for retraining and re-bucketing. The full work is divided into various sections and each section is discussed in

A. TCGAN (Two Contexture GAN)

For data preparation we have followed various popular methods for conversions of sentences into numerical values like 1 hot encoding and etc. we will not go into these details as the main work of ours is the model and its firing methods is to which convert a GAN into a TC-GAN. Below is the Figure-2 which has a GAN and added with two contextures one is added between the loop of generator and discriminator and one is after the generator. The mathematical representation of the GAN is also listed below.

$$\text{Generative adversarial network} = \max_G \left(\min_D E(G,D) \right)$$

$$E(G,D) = \frac{1}{2}\mathbb{E}_{x\sim p_t}[1 - D(x)] + \frac{1}{2}\mathbb{E}_{z\sim p_z}[D(G(z))]$$
$$= \frac{1}{2}\left(\mathbb{E}_{x\sim p_t}[1 - D(x)] + \mathbb{E}_{x\sim p_g}[D(x)]\right)$$

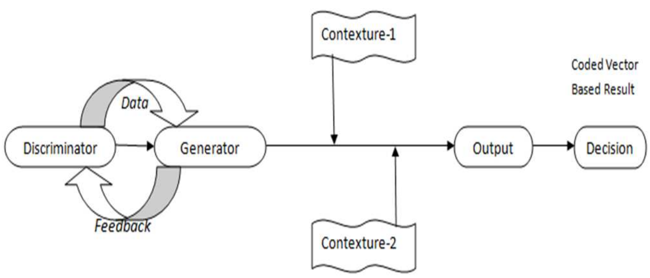

Figure 2: Two Contexture Generative Adversarial Network (TC-GAN)

The mathematical representation of the TC-GAN is as below.

TC-GAN = Quan [min max [log(D(x)) + log(1-D(G(z)))] + error(P1|P2)]

P1 and P2 are the noise data input and Quant is the bucketed output which will have to fall in fixed buckets as per the CNN output layers. This way we will keep training and keep changing the output based on the error functions.

B. Tensor-flow to tenflow-lite

Once we are clear from both sections i.e. Training and result of the model then we ready to develop a lite version for it and make the vocab of corpus and labels with same dimensions along with it. In this model transformation process is required for chat bots has applications in a different field like Q/A for banking field or their apps and for other similar process. Then our next task to know the general views of public and their reaction by this we can get the idea about the system's acceptability in to the market. Tensor-flow-lite is mainly support limited operations or not deal with flexibility of operation of the base tensor-flow and this system mainly deals with the CNN only and does not supports the RNN. So, we are making the python program by which the observed input and output dimensions of our main model and then use tensor-flow method for describing our model and summarized all changes in a process flow.

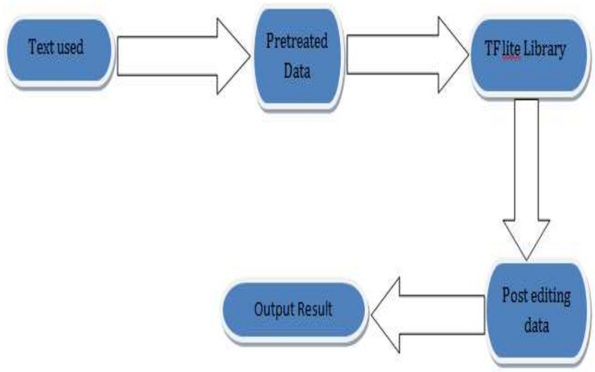

Figure 3: The bucketed model used in Android application using tensor flow lite.

C. Using Tensor-flow-lite model in Android Application.

This process need coding for android app so that it can be used model and take the user output as well as take the user response and reactions submit to the server which is very helpful for future work on it like further analysis on data and training of model and we have source code also for researchers doing their work in context of text generation and future expansion of this project, so that it can be used as a structure for examine their model but consider it as a future work. The biggest limitation of TensorFlow lite is that it takes only CNN not other mathematical functions which are in our models. So, the conversion into CNN from our model is the main task along with keeping the layers of the model intact.

D. Contexture detail

Mainly these two contextures are different structure and layer organized in periodic manner but counting of neurons and input length and vectors comes out are different according to the size of different batches entered. In our model contexture have 3 layers mainly where first layer is about Inverse Conv 1D for help converge the input data into the output vector and categorized. The approximated output of the context is calculated using vectors by this model and through Eigen vector distance measurement method the output method is calculated then which are very useful for feedback for training and comparison.

E. Results of the training from Tensor board

After the completion of training and contextures of the model we have plotted various outputs obtained at different training stages to see the clustering of results. From this plotted data it has been summarized that each cluster has different context. For plotting this cluster, we are using tensor board and it in build tool which shows the result and training checkpoints.

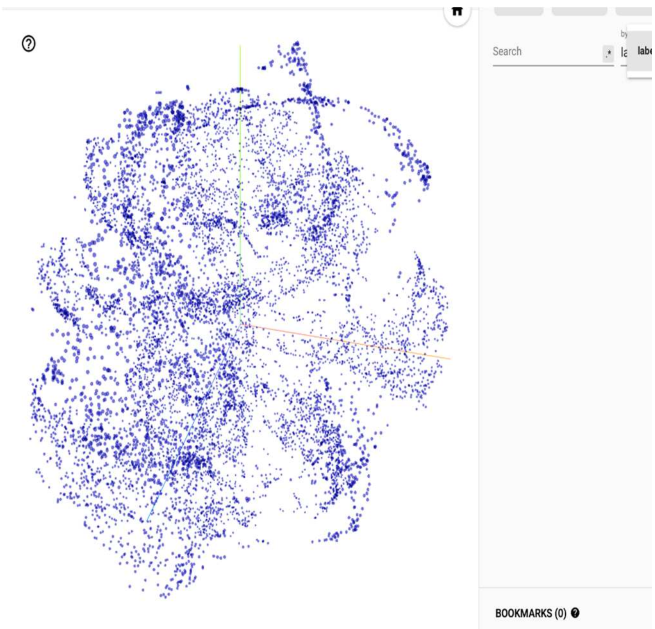

Figure 4: Bucketed results after 100k Iterations

the above is the link for the application, we have build in this paper and available for public use and evaluation which can be used to tailor a new models for the chat bot of any kind as long as it supports the TensorFlow-lite.

Table I: Context and buckets

S.no	No. of Integrations	No. of buckets	Errors	Avg. Context distance
1	10000	12	0.3452	1.3098
2	50000	56	0.21	1.2
3	100000	106	0.11	0.9

III. COMPARISON

Below we want to compare our model with previous Generative adversarial Network (GAN) those work or content are related to our model in a different manner. But as we know previous GAN models have different structure so it is quite difficult to compare all parameters takes in a comparison section instead of we have compared our model with different GAN models as we discuss below. We have used the Large Moview review dataset initially and then user inputs are collected for retrining.

Table II: Comparison with IRGAN

	Training set	**Observation method**	**Arithmetic detail**
IRGAN	Upto 10000 words	Minimax Game	Good
TC-GAN(Our Model)	More Than 10000 words	Context Based	Moderate

Table III: Structural Comparison

	Hidden Layer	**Architectural Concept**	**Configuration**
TexyGAN	Fully connected	Vector Based	Multi-dimensional evaluation
TC-GAN (Our Model)	Embedded	Contextually Based	Context oriented
SeqGAN	Embedded	Probabilistic computation	Contrastive divergence

Table IV: Context Comparison

	Baseline Model	**Model Technique**	**Real and synthetic Data Experiments**
SeqGAN	Binary Classification	Teacher forcing	Good
TextGAN	Feature Matching	MMD loss	Good
GSGAN	Softmax Matrices	Gumbel softmax trick	Moderate
TC-GAN (Our Model)	Contexture Based	Compile based Output	Very effective
MaliGAN	Gradient Saturation	Rescale the rewards	Effective

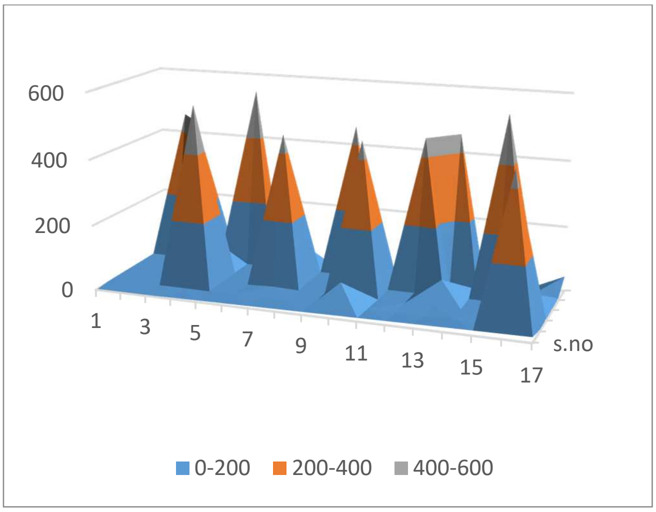

Figure 4: Sample Buckets and samples divided into the sizes (Types of buckets and number of samples in it)

IV Conclusion & Future Work

The TC-GAN model proposed above has bucketed 10k sentences and the as the figures the its categories are getting better over time. The potential of this model is immense as with more input data from users from the android app is getting collected and getting saved for future training. The source code for the same is also available on the GitHub for researchers as well as scholars. We demonstrated that our model can synthesize many bucketed texts of a given source text. Furthermore, our work has shown the potential for generating paraphrase and semantic generation of text. our

References

[1] Dethlefs N and cuayhuitl.H, *Hierarchical reinforcement learning for adaptive text generation. Proceeding of the 6th international conference*, 2010.

[2] Mauldin M.L. Semantic rule based text generation. *Proceeding of the computational linguistias and 22nd annual meeting on association for computational linguistics*, 1984.

[3] Kaiming He, Xiangyu, Shaoing Ren and Jian sun. *Deep residual learning for image recognition, Proceeding of IEEE conference on computer vision and pattern recognition*, 2016.

[4] Yoshua Bengio, Jerome louradour, Ronan cellobert and Jason Weston, *Curriculam learning, Proceedings of the 26th annual international conference on machine learning*, 2009.

[5] Niladri Chatterjee, Anish Johnson and Madhav Krishna, *Some improvement over the blue metric for measuring translation quality for hindi in computing theory and application* , ICCTA-07 International conference on IEEE, 2007.

[6] Lei Yu, Xiang Long, Chao Tong. *Adversarial Control Automation and Artificial Intelligence. Proceeding of the International conference*, 2018.

[7] Wooho Lee, BongNam Noh, Yeosukim ki moon Jeong, Internet and distributed computing system, Proceeding International conference 2019.

[8] Zhiwa Huang, Danda Pani Paudel, Guanju Li, Jiqing Wu, Radu Timofte, Luc Van Gool, Image and video Processing, Proceeding on International Conference, 2019.

[9] Dejia Xu, Vihao chu, Qingyan sun, Computer vision and pattern Recognition, Proceeding on IEEE conference, 2020.

[10] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu, SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient.. In AAAI, 2017.

[11] Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song,and Yoshua Bengio. 2017. Maximum-Likelihood Augmented Discrete GenerativeAdversarial Networks. arXiv preprint arXiv:1702.07983 (2017).

[12] William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: *Better text generation via filling in the*. arXiv preprint arXiv:1801.07736, 2018.

[13] Anirudh Goyal, Nan Rosemary Ke, Alex Lamb, R Devon Hjelm, Chris Pal, Joelle Pineau,and Yoshua Bengio. *Actual: Actor-critic under adversarial learning*. arXiv preprint arXiv:1711.04755, 2017.

[14] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain questionanswering. In EMNLP. Citeseer,2015.

[15] Jun Wang, Lantao Yu, Weinan Zhang*, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, Dell Zhang. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. arXiv:1705.10513v2 [cs.IR] , 2018

[16] Xu, F., G. Harten, D.J. Diner, A.B. Davis, F.C. Seidel, B. Rheingans, M. Tosca, M.D. Alexandrov, B. Cairns, R.A. Ferrare, S.P. Burton, M.A. Fenn, C.A. Hostetler, R. Wood, and J. Redemann, : Coupled retrieval of liquid water cloud and above-cloud aerosol properties using the Airborne Multiangle SpectroPolarimetric Imager (AirMSPI). 2018.

[17] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling arXiv:1412.3555, 2014.

[18] Jonathan Berant, Andrew Chou, Roy Frostig, Percy Liang, Semantic Parsing on Freebase from QuestionAnswer Pairs, EMNLP, 2013.

[19] Mark L. A. Richardson, Emily M. Levesque, AS AN IONIZATION PARAMETER DIAGNOSTIC IN STARFORMING GALAXIES, the American Astronomical Society, 2013.

[20] Pranav Rajpurkar and Jian Zhang and Konstantin Lopyrev and Percy Liang, SQuAD: 100,000+ Questions for Machine Comprehension of Text, arXiv: 1606.05250v3, 2016.

[21] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, Jason Yosinski, Plug & Play

Generative Networks: Conditional Iterative Generation of Images in Latent Space, arXiv:1612.00005v2, 2016

[22] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, Kaheer Suleman, NewsQA: A Machine Comprehension Dataset, arXiv:1611.09830v3, 2016.

[23] Yaroslav Ganin, Victor lempitsky, Unsupervised Domain Adaption by Backpropagation arXiv:1409.7495v2, 2014..

[24] Akos K´ad´ar, Grzegorz Chrupała, and Afra Alishahi. Representation of linguistic form and function in recurrent neural networks. arXiv preprint arXiv:1602.08952, 2016.

[25] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In International Conference on Artificial Intelligence and Statistics, pp. 315–323, 2011.

[26] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. arXiv preprint arXiv:1506.01057, 2015

[27] Kuhnert, N., Lindenmayr, O., & Maier, A. (2017, May). Classification of body regions based on MRI log files. In International Conference on Computer Recognition Systems (pp. 102-109). Springer, Cham

[28] Maier, A., Steidl, S., Christlein, V., & Hornegger, J. (Eds.). (2018). Medical Imaging Systems: An Introductory Guide (Vol. 11111). Springer.

[29] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to PMI-based word embeddings. Transaction of Association for Computational Linguistics, 2016.

[30] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2015.

[31] Adi, Y., Kermany, E., Belinkov,Y .,Lavi,O.,andGoldberg, Y. (2017). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In Proceedings of ICLR Conference Track, Toulon, France.

[32] Chiu, B., Korhonen, A., and Pyysalo, S. (2016). Intrinsic evaluation of word vectors fails to predict extrinsic performance. InFirstWorkshoponEvaluatingVectorSpace Representations for NLP

[33] Prashant kaushik and sunil prasad. Context Aware GAN for sequence text generation in tensor-flow lite for android AI chat application. International Journal of Scientific & Engineering Research Volume 12, Issue 9, September-2021

[34] Kaushik, P. and Yadav, P.K., 2018, August. A Novel approach for detecting malware in Android applications using Deep learning. In 2018 Eleventh International Conference on Contemporary Computing (IC3) (pp. 1--4). IEEE. DOI=https://doi.org/10.1109/IC3.2018.8530668

[35] https://github.com/prashant343/DollyAndroid