Context Understanding

Karan Dhingra (MT19025), Lakshya (MT19067), Snehil (MT19046)

December 10, 2020

1 Abstract

Concept of context Understanding in Natural Language Processing (NLP) has led to paradigm shift from word embedding based models to inculcation of semantics and context based features.It has great applications when it comes to machine language generation,dialog modeling ,etc for improving the quality and relevance.We aim to understand explicit context in terms of three logical relationships between sentences which are contradiction,entailment and neutral.We analyse the effectiveness of current state of art NLP models in capturing the semantics of natural language text in detecting the logical relationship.We propose WC-GANS based approach for generation sentences that maintains the logical relationship given input sentence.

2 Introduction

Recent Deep Learning models like LSTM, transformers, Bert,etc have been very effective in learning representation of languages.But apart from capturing features to preserve the semantics and syntax these models are only able to only use simple context based features (like words embedding,characters,etc).In recent research, use of external context information has shown to improve the models efficiency especially when it comes to task of Natural Language understanding. The ability to determine the logical relationship between sentence pairs is essential to improve the performance of many natural language processing tasks like question answering,text summarization , machine translation,etc. In this research project we focus on recognising and understanding the context whether sentence pairs are entailment, contradiction or independent(neutral) relationship. We try to explore the question of the ability to recognise the logical relationship between two natural language text and further be able to generate sentences that satisfy a given context. The structure of data used for this research consists of sentence pairs that are neutral, contradictory or in an entailment relationship where the second sentence is termed as hypothesis that is said to be inferred from the first sentence that is the premise.The relationship between the two are available in the form of label. At first we explore the effectiveness of state of art models in capturing the semantics to recognise these relationship on 6 different data set of similar structure. We propose are own architecture based on use of context sensitive GANs (Generative adversarial Networks) and GAN architecture similar to that used in images in recent research for the purpose generating the premise given the hypothesis and context(label).

3 Related work

The most recent work for context understanding for the prediction of context given sentence pair by Zhang et al.[1] and Liu et al. 2019[2] using SemBert and Multi-Task Deep Neural Networks (MT-DNN) have given state of the art models for the task. We are majorly interested in generating natural language text that satisfies the given context class.Researchers have explored similar problem but the work was limited to single class that is generating sentences from the entailment class or focused on generating a hypothesis given the premise.Research by (Subramanium et. al. 2018) [3] used conditional GANs using GRU for compression in encoder.They propose to input conditional information that is the sentence and the label by encoding but in contradiction focuses on generating premise given hypothesis.It claims better performance than previous research by (Shen et. al. 2018)[4] which first proposed model to generate sentences conditioned on an logical label (Contradiction, Neutral, Entailment).They do this by modelling sentences as a distribution over latent space that is by using adversarial training.The Architecture for adversarial model consists of (i) Auto-encoder (ii) Prior (iii) Classifier (iv) Discriminator architecture .It proposed novel approach Memory Operation Selection Module(MOSM) as alternative for max pooling. In this paper focus on going generating a premise from a hypothesis and a context label. In order to produce sentences satifying the given context we build two architectures (i) imposing a distribution over an intermediate distribution representing the semantic space of the premise sentence as suggested by (Shen et. al. 2018)[4] (ii) redefine the problem of generating text, which is discrete in nature and limits the generator's training based on discriminator loss. We instead aim to learn an embedding using GAN, which is then decoded by an autoencoder to get sentences as suggested by [5].

4 Methodology

4.1 Model Analysis for prediction of context

Applied existing available models Bert,Roberta,Albert on datasets like RTE, MRPC and WLNI which have similar structure as described before.

4.2 Context Sensitive GAN

The Architecture of the model is similar to that in figure 1. Initially trained Autoencoder-decoder 2 layered bidirectional LSTM on premise to get z representation.The prior uses MLP to create z' embedding from concatenation of hypothesis representation from encoder,label and noise.The Classifier also has encoder and MLP architecture to predict label given z and representation of Hypothesis from architecture similar to prior. Both Classifier and prior are trained and encoder is updated by back-propagation.Finally Discriminator is simple MLP that tries to classify whether embedding is from prior or encoder.First We train discriminator keeping weights of prior and encoder same.Then we Update prior and encoder weights keeping weights of discriminator.The aim is to train prior such that when its embedding is given as input to decoder generates sentence similar to original premise.The model failed to give any good results so we moved on to the next approach.

4.3 TextGAN

In GAN generator relies on the gradients from the discriminator, which flows freely in Images' case as we can represent images in a continuous form to improve. We are limited in a discrete distribution for text, which does not allow the gradient to backflow from discriminator to generator. There are three ways which allow proper training of Text GANs, (i) using Reinforcement learning, (ii) We take argmax on the last layer of the generator(which is non-differentiable and thus continuous so approximation to the argmax were proposed). Finally, [5] (iii) remodeled the problem of generating text to of generating embedding.

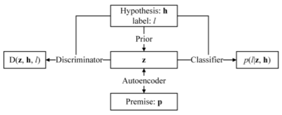

Figure 1: Model comparison on prediction of context given sentence pair Datasets [4]

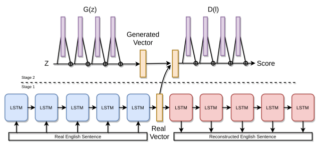

Figure 2: AEGAN : Architecture

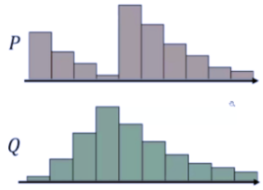

Figure 3: WCGAN loss : Earth Mover's distance between P and Q

TextGan consists of two components, autoencoder, and embed-gan (Fig:2). We pretrain the autoencoder on both premises and hypothesis sentences present in the dataset. It consists of 3 1D-convolutional layers, which aims to learn 5-gram representation of the input data. GRU Network processes the embedding from the convolutional layers to generate encoder features. We only keep the hidden state of the encoder's GRU (latent vector) as embed-gan cannot learn the temporal relationships. The decoder works in an inverse symmetric fashion to the encoder, which the only exception that the GRU network in the decoder operates in an auto-regressive manner. The GAN network is made up of a symmetric discriminator and generator. It consists of three Linear layers followed by LeakyRelu activation function and dropout in a residual fashion. Our initial implementation utilized the loss function of DCGAN[6], which uses binary cross-entropy to classify generator output as fake or true when we minimize discriminator and the generator's loss. As we trained our initial model, we observe the sample's generated from the model had very similar (or same) words, which is known as the problem of Mode Collapse in GAN's. WCGAN [7] aims to solve this problem by replacing the discriminator with critic, as it no longer tries to classify the samples as right or wrong but gives us how far the sample is from our target distribution. Just like in earth mover's distance (fig :3), the GAN's works in similar fashion as the aim is not put discriminator and generator in nash's equilibrium but to make the generator distribution (Q) as close to the target distribution (P). Thus, we use MSE as the loss function for both discriminator and generator.

5 Results

5.1 Context Classification Analysis

Roberta Out performs models in the case of MRPC.For WLNI and RTI both Roberta and Bert have similar performance.

	RTE	MRPC	WNLI
	2.5K	3.7k	721
Roberta	0.69	0.90	0.56
Albert	0.689	0.86	0.52
Bert	0.69	0.83	0.56

Figure 4: Model comparison on prediction of context given sentence pair Datasets

5.2 TexGAN

For Autoencoder, we developed mean model as a baseline, which does not have convolutional layers in the autoencoder and does not decoder encoder's output in autoregressive fashion. We did not develop any baseline for the Generative network, because we believe the current architecture only serves as a baseline. The baseline model got saturated at 0.1 softmax cross entropy loss with the classification accuracy of 0.65, while the ConvAutoEncoder achieved the classification accuracy of 0.92 when both of the models were trained on approximately 1.7l sentences (figures in the appendix).

Table 1: Output of AutoEncoder

ID	Type	Sentence
Sample - 1	Input Output	section covers the relative burden of uso . one one the cost of the uso .
Sample - 2	Input Output	dow chemical will buy rival union carbide dow chemical were to to since carbide .
Sample - 3	Input Output	he believes that trauma centre should get more funding for non mds he believes that trauma centre should get more funding for non mds

The first two outputs in table 5.2 are from the baseline model, while the remaining two from the ConvAutoEncoder. The Sample- 1 belongs to the hypothesis set of the dataset, while Sample- 2 belongs to the premise set, thus highlighting even though the network had been trained on a large dataset of premise samples, it is not able to generalize well to the hypothesis samples. Thus, for GAN model testing we pretrained the ConvAutoEncoder on both premise and hypothesis data samples.

As, we can see most of the sentence generated have analysis, and reached in common. This happened because our generator was not able to learn rich representation of the target distribution(the mode collapse problem). In this experiment, we used our embeddings from our baseline encoder. Thus in order to improve the quality of generation, we changed both the embedding to ConvAutoEncoder, and GAN architecture to generator-critic [7].

Table 2: Output of TexGAN

ID	Type	Sentence
Sample - 1	Input Condition Output	Brilliant, because it fits Neutral she jon growled the . .
Sample - 1	Input Condition Output	One day it is possible we achieve that Entailment he worried happy identifying care it is people have.
Sample - 1	Input Condition Output	she did not look at me nor did she want me to come to her Contradiction I should know outraged i believe it would we did somebody doubtless nothing leave back freeze.
Sample - 1	Input Condition Output	I think they like fort worth . Neutral It about at bush was are nothing over few not another like each section u unless

The network, was able to generate conditional output in some scenarios but it failed to generate meaning in many instances, which is not possible to quantify without human evaluation. In order to analyse the possible cause, we plot the distribution(fig:5.2) and observe that even though the network is not collapsing to few samples. Our network is still not able to generate the complete distribution of the target samples.

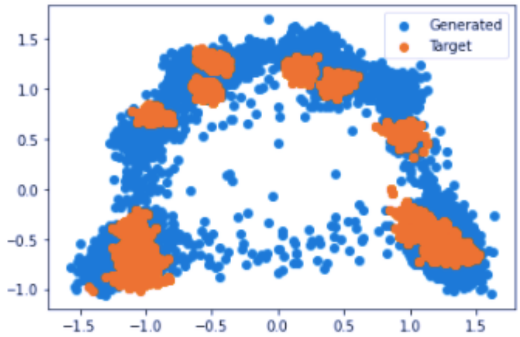

Figure 5: PCA plot of Target and Generated Distribution

6 Individual member contribution

All the team members researched in parallel to understand existing research in the field of context understanding.

6.1 Karan Dhingra (MT19025)

Worked on context based generation Task.Implementing Context sensitive sentence generation using WCGAN's.

6.2 Lakshya (MT19067)

Worked on the initial understanding of finding context between sentence pairs.Analysis on different dataset and Latest research on State Of The Art models.

6.3 Snehil (MT19046)

Worked on context based generation Task. Implementing Context sensitive GAN architecture(RNN Seq2Seq) (i)Autoencoder (ii)classifier (iii) prior (iv)Discriminator

7 Novelty

7.1 In terms of model Architecture

Designed and Implemented Improved GAN architecture for Conditional Text Generation. Implemented and analysed why Seq2Seq architecture would fail in this scenario based on poor results in first proposed architecture.

7.2 In terms of using the dataset for the task at hand

To the best of our knowledge no other work exists in generating contradictory,neutral and entailment sentences on GLUE MNLI dataset.

References

[1] Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, and X. Zhou, "Semantics-aware bert for language understanding," *arXiv preprint arXiv:1909.02209*, 2019.

[2] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," *arXiv preprint arXiv:1901.11504*, 2019.

[3] S. Subramanian, S. R. Mudumba, A. Sordoni, A. Trischler, A. C. Courville, and C. Pal, "Towards text generation with adversarially learned neural outlines," in *Advances in Neural Information Processing Systems*, 2018, pp. 7551–7563.

[4] Y. Shen, S. Tan, C.-W. Huang, and A. Courville, "Generating contradictory, neutral, and entailing sentences," *arXiv preprint arXiv:1803.02710*, 2018.

[5] D. Donahue and A. Rumshisky, "Adversarial text generation without reinforcement learning," *arXiv preprint arXiv:1810.06640*, 2018.

[6] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[7] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," 2017.

8 Appendix

8.1 TexGan

Loss function, and Classification plot for both autoencoder, where x axis denotes time in (hrs).

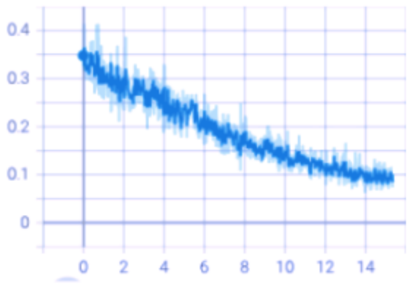

Figure 6: Baseline AutoEncoder - Loss function

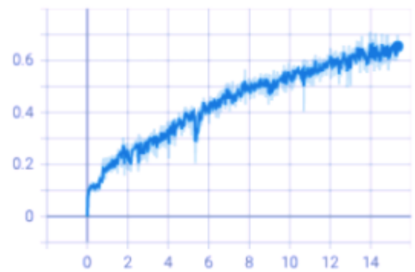

Figure 7: Baseline AutoEncoder - Accuracy Plot

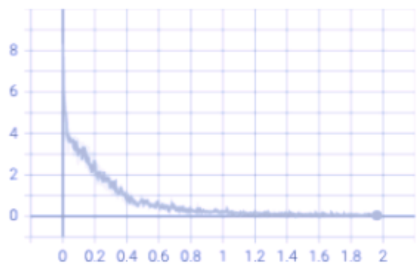

Figure 8: Conv AutoEncoder - Loss function

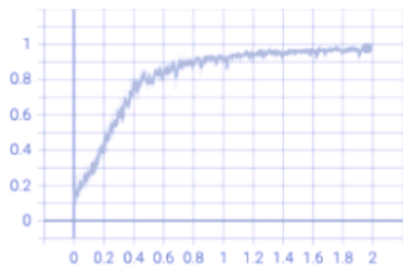

Figure 9: Conv AutoEncoder - Accuracy Plot