DREU Final Report

Sophia Hager
Advised by Nathan Dennler

August 2021

1 Introduction

Text generation is an important task for Natural Language Processing, as it provides a foundation for other NLP applications. Unfortunately, a pitfall of the current approaches towards text generation is that it creates similar or identical sentences[8]. The frequency-based training system prioritizes generation of similar phrases- for example, roBERTa, which produces state-of-the-art results on next-sentence prediction and fill-in-the-blanks tasks [3] chooses words explicitly by how likely they are to fit the context, so their generated sentences are unlikely to be surprising or unexpected, and will likely fit a pattern. The problem we attempted to solve in this project was generating sentences that vary along certain variables, such as length and sentiment.

To solve this problem, we used quality diversity algorithms to attempt to explore the latent space of a sequence to sequence generation system. Because a sequence to sequence Long Short Term Memory network (LSTM network) receives a vector that could be produced by a quality diversity system, quality-diversity algorithms can be used to create text, a previously under-explored application of quality diversity algorithms. Due to the lack of a good evaluation function for how likely a sentence is to be similar to a human sentence, a secondary objective was to create an objective function that assessed and scored sentences' similarity to existing sentences.

2 Relevant Literature

Fontaine et al.[1] describe the problem of *latent space illumination*, or LSI, where given an objective function and additional functions that measure variables of generated content, the goal is to generate a collection of objects that span all possible combinations of the additional functions while maximizing the objective function. They approach this problem with *quality diversity algorithms* (QD algorithms), a class of algorithms described by Pugh et al.[4] that, instead of attempting to find one "best" solution, aim to find a variety of solutions that differ across specified attributes.

Pyribs[7] is a python library that describes itself as "a bare-bones Python library for quality diversity optimization." It utilizes a QD algorithm called Covariance Matrix Adaptation MAP-Elites[2] which has shown significant potential in comparison to other QD algorithms. Pyribs opens up the ability to use QD algorithms on LSI tasks more easily and apply it to less commonly explored tasks, such as text generation.

However, evaluation of QD algorithms can be complicated, since not only can the quality be evaluated, but it should also take into account the variety of solutions created. Pugh et al. [5] suggests using the sum of the fitness of the solution over every satisfied combination of variables. This approach gives weight both to maximizing the fitness of each solution and to generating a large variety of solutions.

3 Methods

To use pyribs [7] on a generation algorithm, pyribs requires the generation network, an objective function, and the behavioral functions that quantify the attributes the network should diversify across. In this case, the generation network was a sequence to sequence LSTM trained on Shen et al.'s Yelp review dataset[6]. This dataset contains approximately 450,000 sentences, each labelled by whether they are from a positive review or a negative review. Each sentence was converted to a sentence embedding vector of size 1024 by roBERTa's sentence embedding constructor [3], then passed to the generator. The generator's attempt to replicate the sentence was evaluated using negative log likelihood loss.

For this project, the objective function's purpose was to evaluate how realistic or likely the sentences that were outputted from the discriminator were. There were no obvious candidates that already existed, so an objective function was constructed. The objective function was trained to differentiate realistic sentences (the sentences from Shen et al.'s dataset [6] and sentences generated by the generator from random numbers that simulated an embedding.

The two variables that the network's outputs were supposed to vary on were the length of the sentence (how many words were created in the sentence) and the sentiment of the sentence (positive reviews versus negative reviews). The length function simply counted the number of words in the sentence using Python's built in split() function. To calculate the sentiment, a classifier was trained on the reviews to give a score between 0 and 1, where 0 was a negative review and 1 was a positive review.

With these constraints satisfied, it was possible to use pyribs to generate random vectors of size 1024 to simulate sentence embeddings. The QD algorithm optimizes the random vectors over time, trying to maximize the objective function, and returns the best-performing options for each combination. Using Pugh et al.'s suggested evaluation method[5], the performance of the QD algorithm can be compared to completely random data.

4 Results

For the text generation project,the goal was to fill 30 "bins" (combinations of variables): there were three ranges for the length to vary between, and ten different ranges for the sentiment to vary between. The total of the fitness of the solutions for 150000 random numbers was 20.44, while the total using the QD algorithm optimization was 30.00. Both filled every bin, but the overall quality of the completely random numbers was on the whole lower.

However, the quality of the sentences in both Figure 1 and Figure 2 shows that the objective function, while reliable on the test set to .993, is not reliable on correctly rating how realistic these sentences are. In particular, each of the sentences in Figure 2 was given a perfect score

	Most Negative Sentiment	Most Positive Sentiment
Short Length	"exexexex times!.."	"i friendlytetete.."
Medium Length	"ok should should be be be to to.."	"andand well well well well well well well well"
Long Length	"this place is actually place place place else else else else else."	"very good good good good good good good good good good are good are are are is is is the is the so so- so----y"

Figure 1: Examples of the best sentences produced by running the generating network on random numbers

	Most Negative Sentiment	Most Positive Sentiment
Short Length	"no environment environment environment environment.."	"beautiful environment environment environment."
Medium Length	"no environment environment to a a environment environment."	"very clean environment environment environment environment environment.."
Long Length	"the environment environment environment would to to to environment."	"very environment environment environment a quality environment and for for for environment environment.."

Figure 2: Examples of the best sentences produced by using pyribs's QD optimization algorithm

by the objective function, when clearly they are not good sentences. Another side effect of the optimization function is that sentences end up looking very similar, with minor tweaks that give the variety for the diversity portion of the algorithm (for example, ”no environment” rather than ”beautiful environment” is enough to sway the classifier that the first is negative and the latter is positive). The random sentences have more variety, but are even more eclectic and less human-readable. The examples suggest that the approach is valid and produces better data than random data, but that it fails due to the flaws with the objective function.

5 Conclusions and Future Directions

It is apparent that the limitations of the objective function hampered the performance of this setup. The results would be more impactful if there were a more reliable objective function that could better discern if a sentence was human-quality or not. To that end, more labelled data would be required to better tell if a sentence is ”good” or ”bad”. Larger scale projects could potentially train a better objective function, but currently, the performance of the QD algorithm is limited by the flaws in the objective function. Ultimately, a better experimental setup may be to use a GAN to generate the sentences, as it fulfills the ability to produce sentences from noise while also training an objective function at the same time.

Another potential direction could be using this setup for sentiment transfer. Using an existing sentence embedding and utilizing the vectors produced by Pyribs as noise, while changing the objective function to evaluate how similar the generated sentence is to the original, there could be a way to preserve the content of the original sentence while varying the sentiment.

References

[1] FONTAINE, M. C., LIU, R., KHALIFA, A., TOGELIUS, J., HOOVER, A. K., AND NIKOLAIDIS, S. Illuminating mario scenes in the latent space of a generative adversarial network. In *AAAI* (2021).

[2] FONTAINE, M. C., TOGELIUS, J., NIKOLAIDIS, S., AND HOOVER, A. K. Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (New York, NY, USA, 2020), GECCO ’20, Association for Computing Machinery, p. 94–102.

[3] LIU, Y., OTT, M., GOYAL, N., DU, J., JOSHI, M., CHEN, D., LEVY, O., LEWIS, M., ZETTLEMOYER, L., AND STOYANOV, V. Roberta: A robustly optimized bert pretraining approach, 2019. cite arxiv:1907.11692.

[4] PUGH, J., SOROS, L., AND STANLEY, K. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI 3* (07 2016).

[5] PUGH, J. K., SOROS, L. B., SZERLIP, P. A., AND STANLEY, K. O. Confronting the challenge of quality diversity. *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (2015).

[6] SHEN, T., LEI, T., BARZILAY, R., AND JAAKKOLA, T. Style transfer from non-parallel text by cross-alignment. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2017), NIPS'17, Curran Associates Inc., p. 6833–6844.

[7] TJANAKA, B., FONTAINE, M. C., ZHANG, Y., SOMMERER, S., DENNLER, N., AND NIKOLAIDIS, S. pyribs: A bare-bones python library for quality diversity optimization. `https://github.com/icaros-usc/pyribs`, 2021.

[8] XU, J., REN, X., LIN, J., AND SUN, X. Diversity-promoting GAN: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, Oct.-Nov. 2018), Association for Computational Linguistics, pp. 3940–3949.