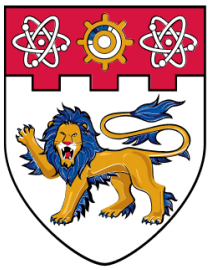


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

NLarge your dataset: Data augmentation for NLP

NTU SC4001 Sem 1 2024

Ng Tze Kean
U2121193J

Gui Zhang Yan Dexter
U2121991C

November 24, 2024

Abstract

In this report, we explore the application of data augmentation (DA) techniques for Natural Language Processing (NLP) using a variety of methods including Large Language Models (LLMs). We demonstrate the effectiveness of these techniques in enhancing the diversity and robustness of the training data, potentially improving the performance of NLP models. We present our methodology, experimental results, and discuss the implications of our findings.

Overall, we found that use of statistical methods such as substitution for data augmentation has limited applications. The use of more advanced deep learning models such as RNN with attention mechanisms tend to perform better in this task. The best results were obtained using Large Language Models (LLMs) for DA, which significantly improved the performance of the model.

1 Introduction

DA is a widely used technique in machine learning to enhance the diversity and robustness of training datasets. By artificially expanding the dataset, DA helps improve the generalization capabilities of models, particularly in scenarios where labeled data is scarce or expensive to obtain [2]. In the context of Natural Language Processing (NLP), DA poses unique challenges due to the complexity and variability of human language.

Traditional DA methods in NLP, such as synonym replacement, random insertion, and back-translation, have shown limited effectiveness in generating diverse and meaningful variations of text data. These methods often fail to capture the nuanced semantics and contextual dependencies inherent in natural language, leading to suboptimal improvements in model performance.

Recent advancements in deep learning, particularly the development of Large Language Models (LLMs) like GPT-2, GPT-3, and T5, have opened new avenues for DA in NLP. These models, pre-trained on vast corpora of text data, possess a remarkable ability to generate coherent and contextually relevant text. Leveraging LLMs for DA involves generating synthetic data samples by providing prompts based on existing training examples.

1.1 Literature Review

DA has been a widely researched area in the field of Natural Language Processing (NLP) due to its potential to enhance the diversity and robustness of training datasets [2]. In the context of sentiment analysis, DA techniques are particularly valuable as they help improve the generalization capabilities of models, especially when labeled data is scarce [4].

Rule based methods like random replacement are quick to implement but lack the generalisability to different corpus. These methods aim to generate new training samples by making small perturbations to the existing data, thereby increasing the size of the training set and improving the generalization capabilities of sentiment analysis models [6].

Interpolation methods such as synonym replacement has also been developed [5] where words in a sentence are replaced with their synonyms. This method has been shown to improve model performance by introducing lexical diversity. However, it often fails to capture the nuanced semantics and contextual dependencies inherent in natural language, leading to suboptimal improvements in sentiment analysis tasks [5].

Leading us to the current state of the art, the use of LLMs for data augmentation has shown promising results in improving the performance of NLP models [1]. By leveraging the generative capabilities of LLMs we are able to reduce the amount of noise introduced and thus generate a higher quality dataset. Most of the research has been focused on NER tasks, and we aim to explore the feasibility of using LLMs for DA in sentiment analysis tasks to ascertain the effectiveness of this approach.

Our hypothesis is that the benefits of LLM DA will still continue to provide superior performance in sentiment analysis tasks over pre-LLM DA methods.

1.2 Objective

In this project, we explore the application of DA techniques for sentiment analysis in NLP. We aim to evaluate the effectiveness of traditional DA methods such as random substitution and synonym replacement, as well as advanced techniques using LLMs for data augmentation.

The outcome would be a python library made open source with the implemented methods and algorithms for the community to use.

1.3 Training and Evaluation

To evaluate the performance gain of the DA techniques, we augmented the dataset at different levels: 5%, 10%, and 20%. Afterwards, we will attempt extreme cases of DA at 50%, 100%, and 200% to observe the trend of the performance.

The base line model will be trained on the original dataset using a Recurrent Neural Network (RNN) model to perform sentiment classification on the validation dataset. We will then retrain the model using the augmented dataset and compare the performance metrics to assess the impact of DA on model performance.

We will also include a comparison of the performance of the RNN model with a Long Short-Term Memory (LSTM) model to evaluate the impact of the model architecture on the effectiveness of DA techniques.

1.3.1 Model Architecture for Sentiment Classification

The RNN model architecture consists of an embedding layer, followed by an RNN layer, and a fully connected layer with a sigmoid activation function for binary classification. The model was trained using the Adam optimizer with a learning rate of `5e-4` and binary cross-entropy loss.

We adopted a pre-trained word embedding model to initialize the embedding layer to transfer knowledge from the pre-trained model to our sentiment classification task. The RNN layer uses a simple hidden layer our target is to train that task layer to perform sentiment classification.

Our primary measure of improvement is the accuracy gain on the validation dataset by the task specific RNN model. We also monitor the loss curves to ensure that the model is not overfitting to the training data. Our hypothesis is that the model will generalize better to unseen data with the augmented dataset through various means to increase the volume of the corpus in meaningful manners that will help the model learn and overfit less to the training data.

2 DA using Random Methods

2.1 Random Swap

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a sequence of words in a text, where x_i represents the i-th word in the sequence.

The random swap process can be defined as follows:

For each word x_i in the sequence X, with a probability p, swap x_i with its adjacent word if x_i is not a stop word. Where $x_j = x_{i+1}$ or $x_j = x_{i-1}$.

$$x'_i = \begin{cases} \text{swap}(x_i, x_j) & \text{with probability } p \\ x_i & \text{with probability } 1 - p \end{cases}$$

where x'_i is the new word after swap. The augmented sequence $X' = \{x'_1, x'_2, \ldots, x'_n\}$ is then used as the new training sample.

The random swap process involves iterating over each word in the sequence and swapping with a predefined probability p. This introduces variability into the dataset, potentially improving the robustness and generalization capabilities of NLP models.

2.2 Random Substitution

The random substitution process can be defined as follows:

For each word x_i in the sequence X, with a probability p, replace x_i with a randomly chosen word from the vocabulary V.

$$x'_i = \begin{cases} \text{random}(V) & \text{with probability } p \\ x_i & \text{with probability } 1 - p \end{cases}$$

where x_i' is the new word after substitution, and $\text{random}(V)$ denotes a randomly selected word from the vocabulary V. The augmented sequence $X' = \{x_1', x_2', \ldots, x_n'\}$ is then used as the new training sample.

The random substitution process introduces variability into the dataset by replacing words with random words from the vocabulary, potentially improving the robustness and generalization capabilities of NLP models.

2.3 Random Deletion

The random deletion process can be defined as follows:

For each word x_i in the sequence X, with a probability p, delete x_i from the sequence.

$$x_i' = \begin{cases} \emptyset & \text{with probability } p \\ x_i & \text{with probability } 1 - p \end{cases}$$

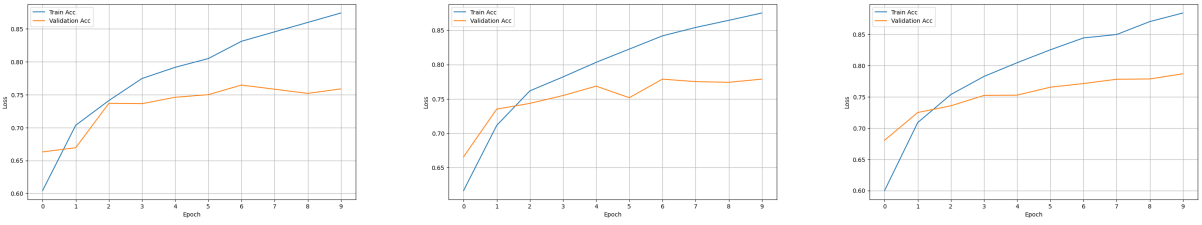
where x_i' is the new word after deletion, and $\emptyset$ denotes the deletion of the word. The augmented sequence $X' = \{x_1', x_2', \ldots, x_n'\}$ is then used as the new training sample.

The random deletion process introduces variability into the dataset by randomly removing words, potentially improving the robustness and generalization capabilities of NLP models.

2.4 Results and Analysis

To evaluate the effectiveness of random methods, we conducted experiments with different levels of augmentation: 5%, 10%, and 20% of the dataset. The performance of the models trained with these augmented datasets was assessed using accuracy score. We will report the results using random swap augmentation as we saw no significant difference in the performance of the model using the different random augmentation methods in practice..

The results of our experiments indicate that the performance of the RNN model keeps increasing with higher levels of augmentation. This suggests that data augmentation provides a clear benefit for sentiment classification tasks. Specifically, the model trained with 20% augmented data achieved the highest accuracy, followed by the models trained with 10% and 5% augmented data. These findings highlight the importance of data augmentation in enhancing the diversity and robustness of training datasets, leading to improved model performance.



(a) 5% Augmentation (b) 10% Augmentation (c) 20% Augmentation

Figure 1: Accuracy graphs for random augmentation DA at 5%, 10%, and 20% levels.

The observed trend, where the performance of the RNN model improves with higher levels of data augmentation, can be attributed to several key factors. Firstly, data augmentation techniques like random swap introduce increased data diversity by exposing the model to a wider range of sentence structures. This diversity helps the model learn more robust representations, enhancing its ability to generalize to unseen examples.

Secondly, data augmentation mitigates over-fitting by effectively increasing the size of the training dataset, reducing the likelihood of the model memorizing specific examples and encouraging it to learn general patterns instead. Additionally, the introduction of variations in the training data makes the model more robust to noise and variations in real-world input data. This robustness is crucial for achieving good performance on unseen data. An example would be a change in the sentence structure from `I love this movie` to `I movie this love`. which can help the model learn patterns that might be found in the human speech patterns and thus generalize better on the test data.

Furthermore, data augmentation acts as a form of regularization, preventing the model from becoming too complex and over-fitting the training data. By providing a more comprehensive training dataset, data augmentation improves the model's generalization capabilities, leading to better performance on validation and test datasets. Collectively, these factors contribute to the model's enhanced ability to learn effectively from the training data and perform better on unseen examples.

3 DA using Synonym Substitution

In this subsection, we explore the performance of DA using synonym substitution.

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a sequence of words in a text, where x_i represents the i-th word in the sequence. Let $S(x_i)$ be the set of synonyms for the word x_i.

The synonym substitution process can be defined as follows:

For each word x_i in the sequence X, with a probability p, replace x_i with a randomly chosen synonym from $S(x_i)$. Mathematically, this can be expressed as:

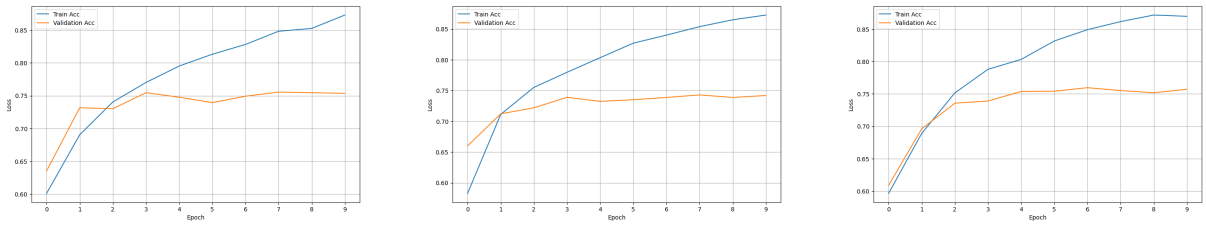
$$x_i' = \begin{cases} \text{random}(S(x_i)) & \text{with probability } p \\ x_i & \text{with probability } 1-p \end{cases}$$

where x_i' is the new word after substitution, and random$(S(x_i))$ denotes a randomly selected synonym from the set $S(x_i)$.

The augmented sequence $X' = \{x_1', x_2', \ldots, x_n'\}$ is then used as the new training sample.

3.1 Results and Analysis

We can see from the results that generally as the augmentation level increases, the performance of the model steadily increases as well.



(a) 5% Augmentation (b) 10% Augmentation (c) 20% Augmentation

Figure 2: Accuracy graphs for synonym augmentation DA at 5%, 10%, and 20% levels.

4 Extreme DA

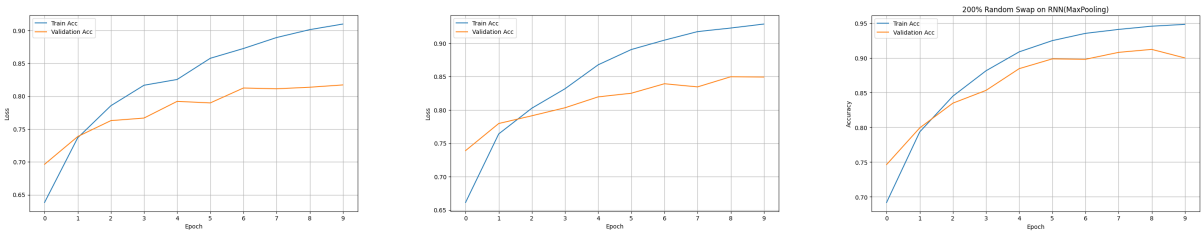
We define extreme DA as augmentation past 50% of the dataset. To explore the trend of the performance, we augmented the dataset at 50%, 100% and 200% levels on each of the above methods and observed the performance of the model.

4.1 Results and Analysis

What is interesting is the continued improvement of the model. The RNN model consistently improves with higher levels of augmentation. This suggests that the RNN model is more robust to the increased data diversity introduced by extreme DA (Figure 3, 4).

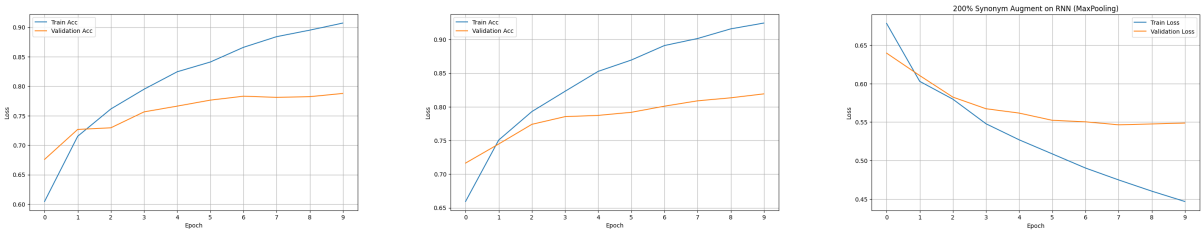
5 DA using Hybrid (Synonym + Random) Substitution

In light of the results from above, we are interested how the performance might change in the case of a hybrid augmentation. We combined the synonym and random and explored the changes in the performance of the model.



(a) 50% Augmentation (b) 100% Augmentation (c) 200% Augmentation

Figure 3: Accuracy graphs for random augmentation DA at 50%, 100%, and 200% levels.

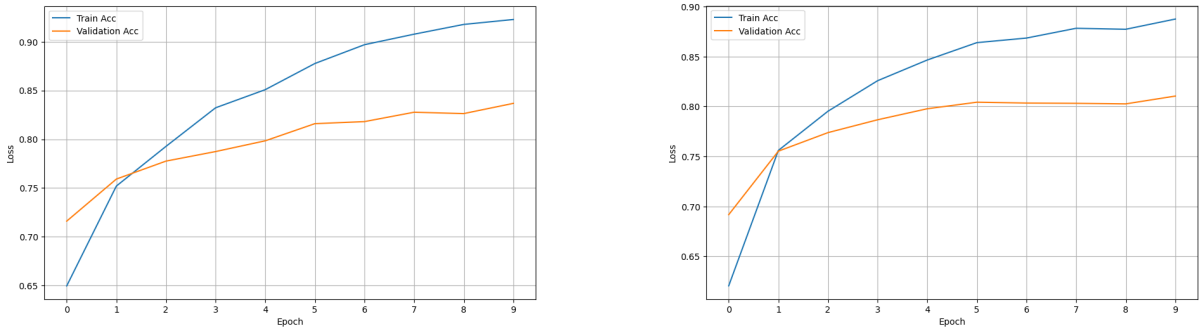


(a) 50% Augmentation (b) 100% Augmentation (c) 200% Augmentation

Figure 4: Accuracy graphs for synonym augmentation DA at 50%, 100%, and 200% levels.

5.1 Results and Analysis

In general, when using the hybrid augmentation, the performance of the model becomes more sporadic and not as consistent as using individual methods alone. When a single method dominates the hybrid augmentation, such as a 100% random + 50% synonym approach, the performance of the model varies depending on the seed used.

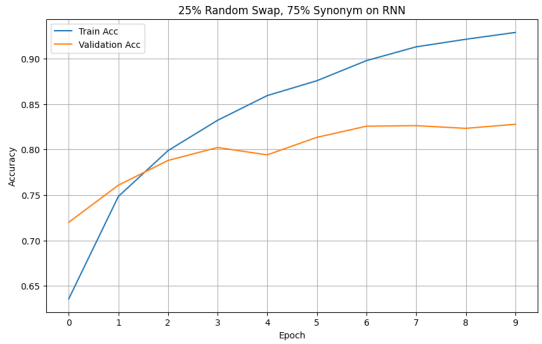


(a) 50% Random + 50% Synonym Augmentation on RNN

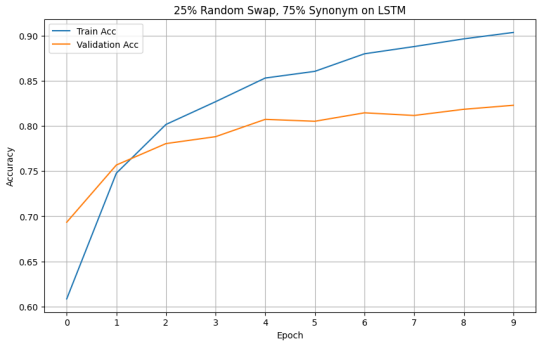
(b) 50% Random + 50% Synonym Augmentation on LSTM

Figure 5: Accuracy graphs for hybrid augmentation levels.

It is worth noting that comparing both Figure 6 and 7 we can see that there little performance gain. In other settings, the performance of the model is less stable and takes more time for convergence.

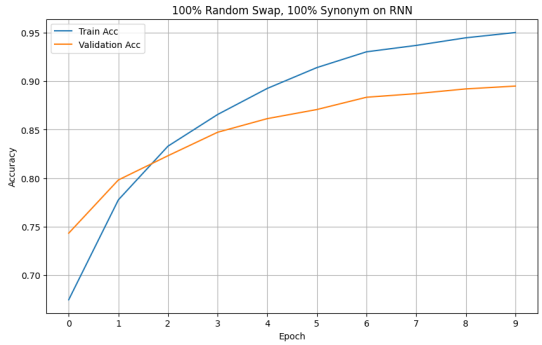


(a) 25% Random + 75% Synonym Augmentation on RNN

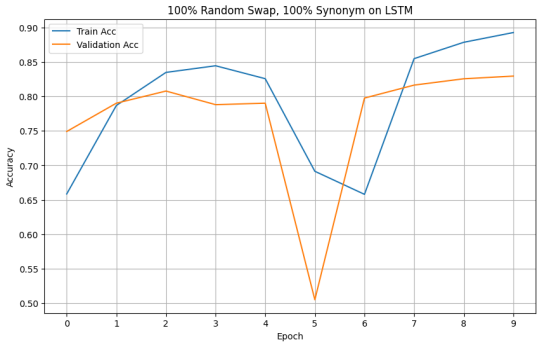


(b) 25% Random + 75% Synonym Augmentation on LSTM

Figure 6: Accuracy graphs for hybrid augmentation levels.



(a) 100% Random + 100% Synonym Augmentation on RNN



(b) 100% Random + 100% Synonym Augmentation on LSTM

Figure 7: Accuracy graphs for hybrid augmentation levels.

This presents an interesting observation that mixing augmentation methods togethers could instead cause degradation of performance or waste compute resources when not tuned properly.

In light of this, we want to explore if the use of LLMs for DA could provide better results.

6 DA using LLMs

To enhance the diversity and robustness of our dataset, we employed a data augmentation technique leveraging pre-trained Large Language Models (LLM) from the Transformers library.

Specifically, we utilized the model to generate synthetic data samples by providing prompts based on existing training examples. We used types of approach to generate the samples. The first is using a model through question and answering to generate the samples. The second is to use a model that performs summarization of content.

The summarizer was easy to implement as it generated relevant samples, however, the question and answer technique was sporadic and often generated repeated text. We will note some of challenges and solution that we applied in this approach.

6.0.1 LLM selection

The choice of the model is crucial as it determines the quality of the generated samples. We experimented with several models, including GPT-2, GPT-3, and T5. As we tried to perform prompt engineering to generate diverse samples, we found that these models could not adequately paraphrase the training data, more often than not, producing the exact same text or slightly modified versions of the original text.

We suspect that these LLM do not perform well due to the lack of contextual information in the prompt. We hypothesize that the models require more contextual information to generate diverse samples. We realized that a model that allows us to specify a role for the prompt, we would be able to generate more diverse samples.

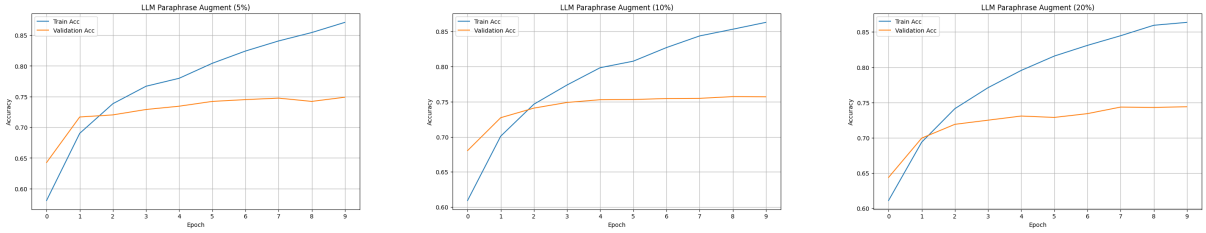
6.0.2 Prompt Engineering

We proceeded to experiment with the Qwen model, with some of the techniques applied from [3]. We found that the Qwen model allows us to specify a role for the prompt, which allows us to instruct the model to specifically only paraphrase the verbs and structure of the sentence. This allows us to generate a text that differs from the original text, while still retaining the meaning of the original text.

6.0.3 Results and Analysis

6.0.3.1 Question and Answer Model

We observed that with the Qwen model, we were able to generate diverse and relevant samples, yet the performance of the model did not improve and in fact there was degradation in the performance of the model. This can be seen from Figure 8 and Table 1. This is very unexpected as we hypothesized that the diverse samples would improve the performance of the model. Our hypothesis that using a expert model to increase the diversity of the samples would improve the performance of the model was not supported by the results.



(a) 5% Augmentation (b) 10% Augmentation (c) 20% Augmentation

Figure 8: Accuracy graphs for LLM substitution DA at 5%, 10%, and 20% levels on RNN.

We examined the possible causes of this unexpected result and found that the augmentation generated by the Qwen model was too long at 100 tokens that the RNN suffered from vanishing gradients. We hypothesize that there was 2 main issues:

1. Vanishing gradients due to the long sequences generated
2. RNN suffering from the long sequences generated

Augmentation Level	Train Accuracy	Validation Accuracy
5%	0.875	0.751
10%	0.873	0.757
20%	0.873	0.748

Table 1: Performance of RNN model with different levels of DA with 100 token per sample limit.

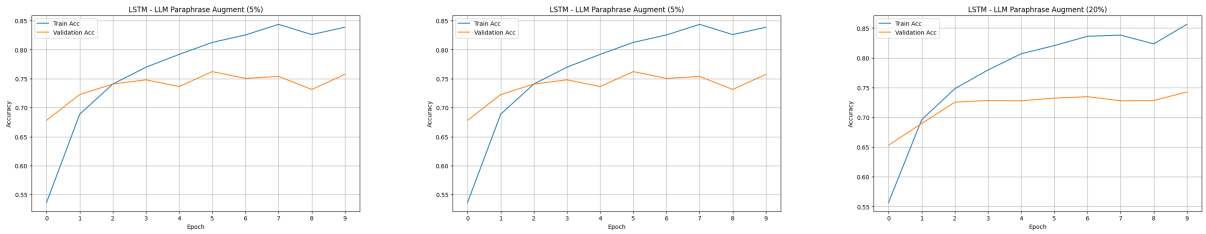
We tailored our approach in 2 ways:

1. We limited the token length of the generated samples to 20 tokens
2. We changed the RNN model to a LSTM model to better handle the sequences that could possibly be too long for the RNN.

It is interesting to see that even on a more complex model, the performance gain of augmentation through LLM is limited or worse than the rule-based methods (Table 9).

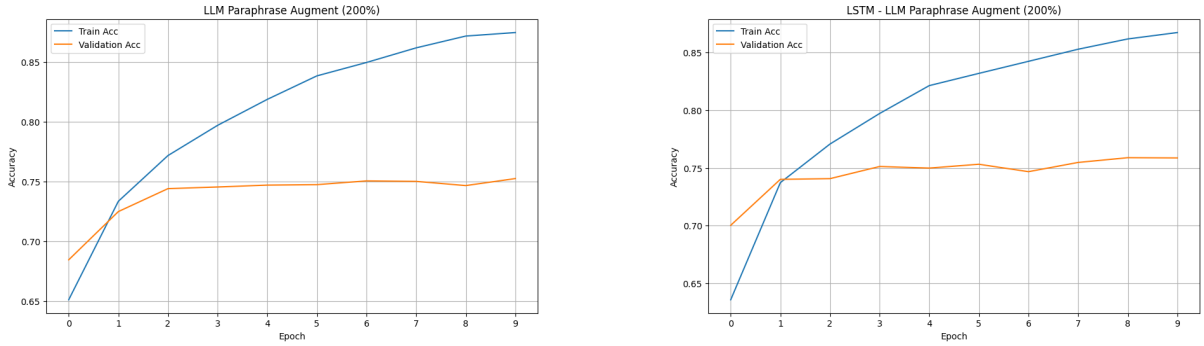
We can also see similar trends in the extreme cases of data augmentation in the LLM setting as well. We further examined the potential causes and narrowed down the issue to the following:

1. The generated samples contained too many OOV that the model was not able to learn from. Our embedding layer only updates parameters for known words and does not learn from OOV words.



(a) 5% Augmentation (b) 10% Augmentation (c) 20% Augmentation

Figure 9: Accuracy graphs for llm substitution DA at 5%, 10%, and 20% levels on LSTM.



(a) 200% LLM Augmentation on RNN (b) 200% LLM Augmentation on LSTM

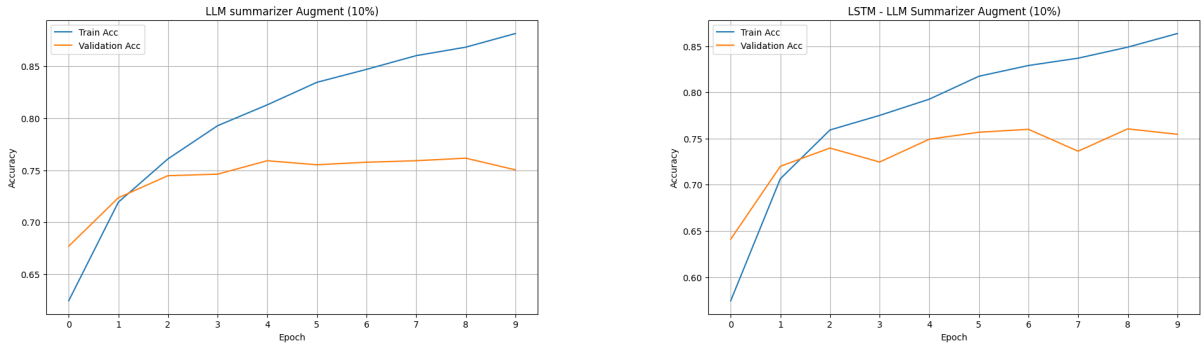
Figure 10: Accuracy graphs on extreme LLM augmentation.

2. The noise introduced by the LLM was too high that the model was not able to learn from the samples.

To handle the first issue, an embedding layer that learns from OOV words and updates its vocab could be used. The handling of the second issue is more complex as it would require a more detailed prompt engineering to generate samples that are more relevant. Instead of moving with that approach, we opted to use a summarization model to generate the samples from the existing data.

6.0.3.2 Summarization Model

The summarizer model performs almost the same as the LLM model in the previous section. The performance of the model (Figure 11) compared to the random and synonym techniques did not show a clear advantage when augmentation is below 20%. The performance gain only starts to show when augmentation levels continued to increase. The summarizer model chosen is the **facebook Bart** model.



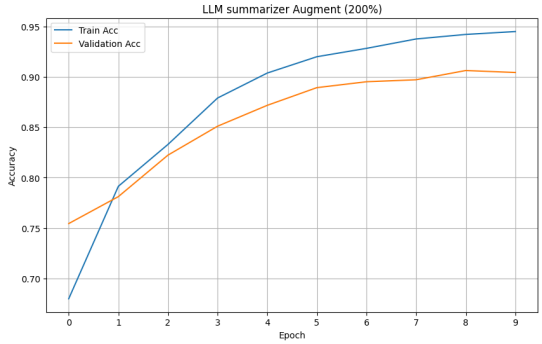
(a) 10% LLM (Summarize) Augmentation on RNN (b) 10% LLM (Summarize) Augmentation on LSTM

Figure 11: Accuracy graphs on LLM (Summarize) augmentation.

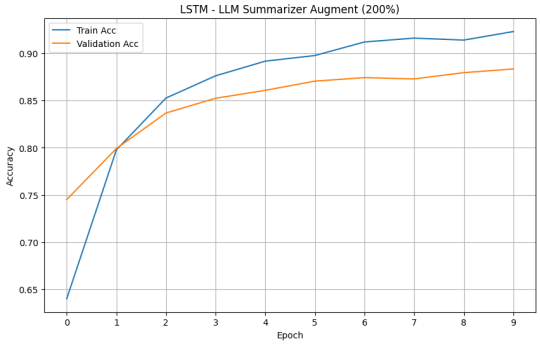
With augmentation levels at 200%, the performance of the model improved significantly (Figure 12). The levels of accuracy of the models exceeded the performance of the rule-based methods by a huge

margin with the RNN model out performing over 90% accuracy. The LSTM model also performed well with over 87% accuracy.

On investigation, we found that the use of summarizers to generate samples produced less OOV counts allowing the model with a static vocab to learn from the samples. We also hypothesize that unlike the task of paraphrasing, the summarizer had better capabilities to capture the essence of the text and generate relevant samples.



(a) 200% LLM (Summarize) Augmentation on RNN



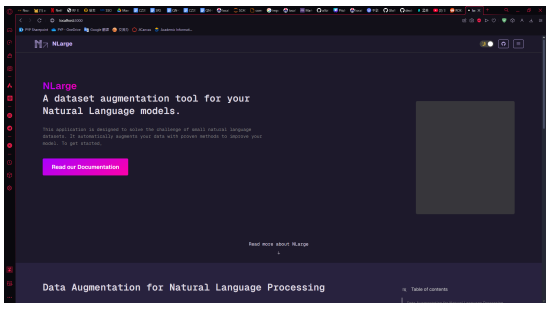
(b) 200% LLM (Summarize) Augmentation on LSTM

Figure 12: Accuracy graphs on LLM (Summarize) extreme augmentation.

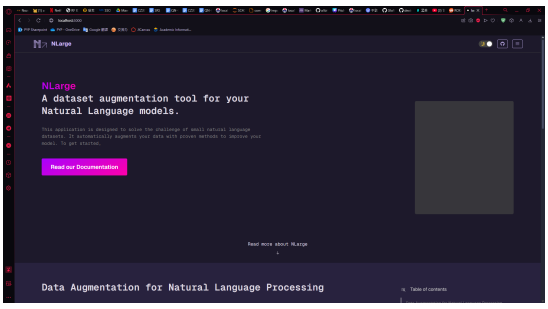
7 NLarge: A Python Library for Data Augmentation

We have developed a Python library called NLarge to DA for NLP sentiment analysis. The library provides the methods and functions used in this report along with models that can be used for sentiment analysis. The goal is to provide a toolkit where users can easily experiment with different DA techniques at various levels and models for sentiment analysis tasks. The library can be accessed at `https://pypi.org/project/NLarge/` and the source code is available at Github.

We also made a website to provide end users with some information `https://nlarge.com` such that we have an end to end solution for users. In the website, we provide documentation on how to use the library and the different methods that are available.



(a) Webpage Home



(b) Webpage Documentation

Figure 13: NLarge website.

7.1 Discussion

The experiments conducted in this study demonstrate the impact of DA techniques on the performance of NLP models for sentiment analysis. Traditional methods such as random substitution and synonym replacement provided a baseline improvement in model performance by introducing variability into the training data. However, these methods often fell short in capturing the nuanced semantics and contextual dependencies inherent in natural language.

The introduction of advanced DA techniques using LLMs marked a substantial improvement over traditional methods. Specifically, the use of LLMs for summarization and paraphrasing generated high-quality augmented data that significantly enhanced model performance. The results indicated that

models trained with LLM-augmented data achieved higher accuracy and better generalization capabilities compared to those trained with traditional DA methods.

The experiments with extreme augmentation levels (e.g., 200%) further highlighted the effectiveness of LLM-based DA. The RNN model, for instance, achieved over 90% accuracy, while the LSTM model reached over 87% accuracy. These findings suggest that LLMs can generate diverse and contextually relevant samples that are beneficial for training robust NLP models.

One key observation was that the use of summarizers to generate augmented samples resulted in fewer OOV counts. This allowed the models with a static vocabulary to learn more effectively from the augmented data. Additionally, summarizers were found to be more capable of capturing the essence of the text and generating relevant samples compared to paraphrasing techniques. The issue of vanishing and exploding gradients in RNNs when dealing with longer sequences could have been countered in practice but this could be mitigated not only by switching to LSTMs or limiting token lengths but also by using techniques such as gradient clipping during training to prevent exploding or vanishing gradients.

Interestingly, the performance of the RNN model was generally better than that of the LSTM model on the Rotten Tomatoes dataset. This can be attributed to several factors. Firstly, the Rotten Tomatoes dataset consists of relatively short text sequences, where the advantages of LSTM's ability to capture long-term dependencies are less pronounced. RNNs, being simpler and less computationally intensive, can perform well on such short sequences without the overhead of managing long-term dependencies.

Secondly, LSTMs are designed to mitigate the vanishing gradient problem in long sequences, but this advantage may not be fully utilized in datasets with shorter text lengths. The additional complexity of LSTMs, including the gating mechanisms, might introduce unnecessary overhead for short text sequences, leading to slightly lower performance compared to RNNs.

We do note that the performance of the LSTM model was not superior to the RNN when the token length of the LLM models were greater than 20 tokens because we did not use a dynamically updating vocabulary size which could explain why in this project, there was no clear advantage of the LSTM model.

We would also like to note that the task chosen at hand is specifically for sentiment classification and that the findings in this report may not translate to other tasks specific domains. However, we do believe that overall the use of DA can potentially help in the case of over-fitting NLP tasks.

Future work can also consider automatic filtering metrics (e.g., perplexity or cosine similarity with original sentences), to ensure the augmented data is both diverse and meaningful when generating samples using LLMs. Additionally, exploring other LLM models and fine-tuning them on specific tasks could provide further improvements in the quality of augmented data.

7.2 Conclusion

In conclusion, this study highlights the importance of data augmentation in enhancing the performance of NLP models for sentiment analysis. Traditional DA methods, while useful, have limitations in capturing the complexity of natural language. Advanced techniques using LLMs, particularly for summarization and paraphrasing, offer a promising solution by generating high-quality augmented data that significantly improves model performance.

The development of the NLarge Python library provides a valuable toolkit for researchers and practitioners to experiment with various DA techniques and models for sentiment analysis. By offering a range of methods and functions, NLarge aims to facilitate the exploration and implementation of effective DA strategies in NLP tasks.

Future work could explore the integration of other advanced DA techniques and further optimization of LLM-based methods. Additionally, expanding the library to support a wider range of NLP tasks beyond sentiment analysis could provide broader applicability and benefit to the NLP community.

Overall, the findings of this study underscore the potential of LLM-based data augmentation to advance the state-of-the-art in NLP model performance and generalization capabilities.

References

[1] Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. Data augmentation using LLMs: Data perspectives, learning paradigms and challenges. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1679–1705, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[2] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. A survey of data augmentation approaches for NLP. *CoRR*, abs/2105.03075, 2021.

[3] Prompt Engineering Guide. Prompt engineering guide: The ultimate guide to generative ai, 2024. Accessed: 2024-10-22.

[4] Zhenhao Li and Lucia Specia. Improving neural machine translation robustness via data augmentation: Beyond back-translation. In Wei Xu, Alan Ritter, Tim Baldwin, and Afshin Rahimi, editors, *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 328–336, Hong Kong, China, November 2019. Association for Computational Linguistics.

[5] Gözde Gül Şahin and Mark Steedman. Data augmentation via dependency tree morphing for low-resource languages. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[6] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China, November 2019. Association for Computational Linguistics.