LLMs vs Established Text Augmentation Techniques for Classification: When do the Benefits Outweight the Costs?

Jan Cegin[♠†], **Jakub Simko**[†], **Peter Brusilovsky**[‡]
[♠] Faculty of Information Technology, Brno University of Technology, Brno, Czechia
[†] Kempelen Institute of Intelligent Technologies, Bratislava, Slovakia
[‡] University of Pittsburgh, Pittsburgh, USA
{jan.cegin jakub.simko}@kinit.sk, peterb@pitt.edu

Abstract

The generative large language models (LLMs) are increasingly being used for data augmentation tasks, where text samples are LLM-paraphrased and then used for classifier fine-tuning. Previous studies have compared LLM-based augmentations with established augmentation techniques, but the results are contradictory: some report the superiority of LLM-based augmentations, while others only marginal increases (and even decreases) in the performance of downstream classifiers. Research that would confirm a clear cost-benefit advantage of LLMs over more established augmentation methods is largely missing. To study if (and when) LLM-based augmentation is advantageous, we compared the effects of recent LLM augmentation methods with established ones on 6 datasets, 3 classifiers, and 2 fine-tuning methods. We also varied the number of seeds and collected samples to better explore the downstream model accuracy space. Finally, we performed a cost-benefit analysis and showed that LLM-based methods are worthy of deployment only when a very small number of seeds is used. Moreover, in many cases, established methods lead to similar or better model accuracies.

1 Introduction

The emergence of recent large language models (LLMs) such as GPT-4, Gemini, Llama, and their wide availability prompted their use in *augmentation* of textual datasets (Ubani et al., 2023; Dai et al., 2023; Piedboeuf and Langlais, 2023; Li et al., 2023; Ding et al., 2023; Cegin et al., 2023, 2024). In most LLM-based augmentation scenarios, the dataset size is increased through paraphrasing of original samples. The extended datasets are then used for training small *downstream* classifiers with small inference costs. LLM augmentation has been used in various domains such as sentiment analysis (Onan, 2023; Piedboeuf and Langlais, 2023), intent classification (Cegin et al., 2023), news classification (Piedboeuf and Langlais, 2023; Cegin et al., 2024) and health symptoms classification (Dai et al., 2023).

While LLM augmentation improves downstream classifiers, it is also costly (power consumption, CO_2 emissions), as generative models often feature parameters in tens of billions. This is magnitudes higher than other *established* (most used) augmentation methods, including *back translation* paraphrasing, or BERT-based *word insertion* and *synonym swap*. Previous works have measured classifier performance, comparing *LLM-based* and *established* augmentation methods. The results have so far been conflicting and mixed with studies reporting the *LLM-based* augmentation to be superior on classifier performance (Ubani et al., 2023; Dai et al., 2023), while others report only marginal gains and even the *established* augmentation outperforming the *LLM-based* one (Piedboeuf and Langlais, 2023). Furthermore, existing studies were limited in terms of parameters, neglecting the variety of available LLMs, the potential impact of the number of seed samples and collected samples, the cost of these methods, and the variety of classifiers and their fine-tuning methods.

The goal of this paper is to compare the accuracy and cost-benefits of the most used *established* text augmentation methods with their recent *LLM-based* counterparts. Compared with previous studies, this paper offers a more systematic and finer-grained comparison over multiple dimensions to specifically identify cases where and how do *LLM-based* augmentation methods outperform *established* text augmentation methods. We are the first to consider the CO_2 emissions and costs of these methods to better identify cases where *LLM-based* augmentation could be preferable. We formulate the following research questions:

RQ1: *Considering downstream classifier accuracy, in which cases do the established tex-*

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 10476–10496
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

tual augmentation methods work equally or better than the LLM-based methods?

RQ2: *In which cases does the cost of using LLM-based textual augmentation methods instead of established ones outweigh its benefits?*

We empirically investigated three techniques commonly used in textual augmentation: *paraphrasing*, *word inserts*, and *word swaps* (replacements). All three exist in both *established* and *LLM-based* variants. In the established variant, paraphrasing is done through back-translation using an RNN (Sennrich et al., 2016), while inserts and swaps use a BERT-based approach (Kobayashi, 2018; Kumar et al., 2020). For LLM-based variants, we prompted 2 LLMs (GPT-3.5 and Llama-3[1]) to perform all three techniques. We experimented with 6 different datasets (with tasks of sentiment analysis, news classification, and intent classification), 3 downstream classifier models (BERT, RoBERTa, DistilBERT), and 2 fine-tuning approaches (fully fine-tuned, and QLoRA (Dettmers et al., 2024)). Furthermore, we investigated various numbers of seeds and collected samples used in the augmentation. Together, this resulted in a total of 267,300 fine-tunings, from which we identified the best-performing LLM and established methods (answering Q1). These were then further scrutinized under cost-benefit analysis (answering Q2).

The most prominent findings are: 1) The best LLM augmentation methods outperform established ones *only* when a small number of seeds is used. The advantage of LLM-based augmentation diminishes with increased seed numbers, making it less cost-feasible. This hints towards using LLM-based methods only in scenarios with a small number of seeds per label (5-20). 2) LLM augmentation methods have a higher impact on the accuracy of less robustly pre-trained classifiers such as DistilBERT or BERT. 3) LLM augmentation methods have a higher impact on classifier accuracy for full fine-tuning when compared to QLoRA fine-tuning.

2 Related Work: Text Augmentation

Text augmentation is a process of increasing the diversity of training text data without necessarily collecting more original (or seed) data. Text augmentation was inspired by image augmentation (Feng et al., 2021; Zhou et al., 2024) where various techniques such as cropping, rotating, flipping, etc. were used to build models that are more robust to image variation and in turn enhance their performance. Text and data augmentation have an increasing number of Google weekly trend searchers in recent years (Anaby-Tavor et al., 2020; Feng et al., 2021; Zhou et al., 2024), indicating an increasing interest in these kinds of model performance enhancing methods.

One of the most established is character-based augmentations (Wei and Zou, 2019; Karimi et al., 2021), where given a seed text, a new sample is created via character insertion, replacement, or deletion. Another method is backtranslation (Sennrich et al., 2016), which translates a given text into one language to then translate it back, essentially creating a paraphrase. Various LLMs such as GPT-2 (Radford et al., 2019) or BART (Lewis et al., 2020) have also previously been used to create paraphrases. Additional extensions used style transfer to create paraphrases of a certain linguistic style (Krishna et al., 2020), syntax control of the generated paraphrases (Goyal and Durrett, 2020; Chen et al., 2020), multi-lingual paraphrases (Thompson and Post, 2020) and LLM fine-tuning using QLoRA for specific domains (Chowdhury et al., 2022). Another established method is the usage of pre-trained LLMs to generate new samples by either word insertion or replacement of words via masking certain parts of the seed text and allowing the LLM to find good replacements for the masked parts of the text (Kobayashi, 2018; Kumar et al., 2020).

Text augmentation methods were adapted with the rise of new LLMs such as GPT-4 or Llama to leverage these new powerful models to generally create paraphrases of given seed texts. A recent study (Piedboeuf and Langlais, 2023) found that the GPT-3.5 paraphrasing provides an increase in classifier accuracy. However, it does not outperform the previously established text augmentation methods by a significant margin (also for low-resource settings). In contrast, two studies reported better performance in using LLMs as data augmenters than using previous state-of-art techniques in both the paraphrasing of existing texts (Dai et al., 2023) and in a zero-shot setup of generating new texts using specific prompts (Ubani et al., 2023) for low-resource settings. A comparison of these studies is problematic, as each study varied a large number of parameters such as different numbers

[1] Albeit BERT is often referred to as an early LLM, for the sake of wording clarity, we do not consider it as such in our study.

of seed samples (from 10 to 1000 per label), different numbers of collected samples per seed used (from 5 to 20), classifiers used, etc. Regardless of the mixed results reported, newer LLMs have been used for a variety of augmentation tasks and domains such as automated scoring (Fang et al., 2023), low-resource language generation (Ghosh et al., 2023), sentiment analysis (Piedboeuf and Langlais, 2023; Ubani et al., 2023; Onan, 2023), news classification (Piedboeuf and Langlais, 2023), content recommendation (Liu et al., 2024) and health symptoms classifications (Dai et al., 2023).

Given the wide usage of LLM augmentation methods and the mixed results of studies (Dai et al., 2023; Piedboeuf and Langlais, 2023; Ubani et al., 2023) comparing them with established augmentation methods, a finer analysis of cases where one of these types of methods is preferable is required.

3 Study Design

To assess the advantages of either *established* or newer *LLM-based* augmentation methods on a finer scale and to tackle the mixed results reported by previous works, we performed a comparative study. At its core was the same basic scenario (see also figure 1): on a given text classification dataset, a number of seed samples were selected for each class. For each seed sample, a given augmentation method generated a number of additional "augmented" samples. Both original data and the augmented samples were then used to fine-tune a downstream classifier. This scenario was repeated for all examined methods and a variety of parameters (see below), resulting in a total of 37,125 augmented samples and 267,300 fine-tunings. Then, the accuracy of the resulting classifiers was compared to answer Q1. To answer Q2, the augmentation costs in terms of computation time, finances, and CO_2 emissions were determined and weighted against accuracy gains in a cost-benefit analysis. We publish all of our measurements, the code, and the data used. [2]

The study had the following parameters:

- the augmentation technique (paraphrasing, contextual word insert, word swap – realized using either established or LLM-based methods),
- the number of seed samples per label (5, 10, 20, 30, 40, 50, 100)[3],
- the number of collected samples per seed (1, 2, 5, 10, 15),
- the LLMs used as augmenters in case of LLM-based methods (GPT-3.5, Llama-3-8B),
- the fine-tuned classifiers (DistilBERT, RoBERTa, BERT),
- the fine-tuning approach (full, QLoRA),
- and the dataset/task (6 datasets).

3.1 Established Text Augmentation Methods

As the *established* text augmentation methods, we chose 3 well-known yet simple and relatively efficient methods as shown by a previous study (Dai et al., 2023). We went with model-based techniques that leverage some form of trained Seq2Seq model or contextual embedding methods that use smaller LLMs (BERT is frequently considered an early LLM). First among them is the *backtranslation* (Sennrich et al., 2016) – in the past (before the advent of LLMs), a popular method used for paraphrasing. The method translates a sentence from one language to another and back to create paraphrases. Another popular and relatively simple method is the replacing or inserting of words based on embeddings. In our experiments, we used contextual embeddings (Kobayashi, 2018; Wu et al., 2019; Kumar et al., 2020). We used two contextual embedding methods: *contextual word insertion* and *contextual word swap* (replacement). As a first step, the *contextual word insertion* method randomly inserts masks between words in a sentence, while the *contextual word swap* method randomly replaces a set number of words in the sentence for masks. Next, a model is queried to get the most likely tokens for each mask. The details of these methods and the parameters used for each of these methods can be found in Appendix I. We used the implementations provided by the NLPAuglibrary (Ma, 2019).

3.2 LLM-based Text Augmentation Methods

As the *LLM-based* text augmentation methods, we implemented the three given techniques using prompts similar to previous works (Cegin et al., 2023; Piedboeuf and Langlais, 2023). However,

[2] Data and code at https://github.com/kinit-sk/llms_vs_nlpaug_data_aug

[3] For some datasets (ATIS, FB), the maximum seed number was lower than 100 due to smaller class sizes, for more details see Appendix H

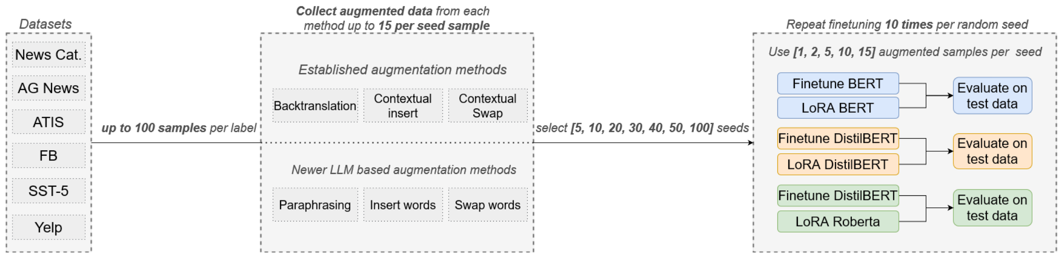
Datasets
News Cat.
AG News
ATIS
FB
SST-5
Yelp
up to 100 samples per label
Collect augmented data from each method up to 15 per seed sample
Established augmentation methods
Backtranslation
Contextual insert
Contextual Swap
Newer LLM based augmentation methods
Paraphrasing
Insert words
Swap words
select [5, 10, 20, 30, 40, 50, 100] seeds
Repeat finetuning 10 times per random seed
Use [1, 2, 5, 10, 15] augmented samples per seed
Finetune BERT
LoRA BERT
Evaluate on test data
Finetune DistilBERT
LoRA DistilBERT
Evaluate on test data
Finetune DistilBERT
LoRA Roberta
Evaluate on test data

Figure 1: Overview of our methodology. For each dataset, we randomly sample 100 samples per label, which are then used to collect up to 15 augmented samples per seed sample. These seeds are then randomly sampled in various sizes and used for fine-tuning with various numbers of augmented samples to evaluate each method.

as the previous works generally used only paraphrasing, we devised new prompts explicitly asking the model to replace words for their synonyms or change the text by inserting words into it. Thus, we had 3 different LLM-based text augmentation methods: *paraphrasing*, where we asked the model to produce a paraphrase, *word insertion*, where we asked the model to produce a new sample by inserting words into seed sample, and *word swap* where we asked the model to produce a new sample by replacing words for their synonyms in the seed sample. We used these 3 methods to gather data using both GPT-3.5 and Llama-3-8B. We opted for not including multiple LLMs due to various versions of LLMs having little effect on classifier performance (Cegin et al., 2024). The inclusion of multiple LLMs would make the study more bloated without much benefit. Details about prompt templates, model types used, and parameters used during inference can be found in Appendix J.

3.3 Datasets

To explore the diversity of augmentation effects, we used 6 different datasets representing three distinct text tasks: the classification of sentiment, intent, and news domains. All datasets were multi-class and English. We used the *News Category* (Misra, 2022; Misra and Grover, 2021) and *AG news* (Zhang et al., 2015) for news classification, *FB* (Schuster et al., 2019) and *ATIS* (Hemphill et al., 1990) for intent classification, and *SST-5* (Socher et al., 2013) and *Yelp* (Zhang et al., 2015) for sentiment classification. When measuring the accuracy of downstream classifiers, we used test splits of each of these datasets. To achieve uniform sizes and distributions, we selected a subset of classes and down-sampled some of them for use in our experiments. Details about the datasets, labels, and class sizes used for each dataset can be found in Appendix H.

3.4 Evaluation Process

For each combination of a number of seeds and datasets, seed samples were randomly selected from among the dataset's classes. Then, the selected augmentation method was applied to generate the additional samples.

We manually checked the *validity* of a random subset (10%) of the collected data (i.e., whether the created samples truly are paraphrases retaining the labels of their seeds). Previous works have already shown that the validity with newer LLM augmentation methods is high (Cegin et al., 2023, 2024), yet we still sought to confirm it and examine the established methods as well. We found the highest validity of samples for the LLM-based *paraphrasing* with 100% valid samples. Both LLM-based and established *word insert* and *word swap* methods achieved 95%-97% validity, struggling mostly with incorrectly named entities. The established paraphrasing *backtranslation* method yields 98%-99% valid samples but also a very large portion of duplicates (around 80%). Details on validity checks can be found in Appendix C.

In terms of the lexical diversity of the collected samples, the insertion types of methods achieved the highest number of unique words in the data and the highest number of unique 4-grams in the data. This was consistently higher than the paraphrasing method in all 6 datasets by about 10 to 20%. The higher diversity, in this case, is easily explainable from the increasing size of the augmented samples produced: in general, the word insertion methods insert words into the original sentence, thus increasing its length by 20-30%, which leads to higher lexical diversity. This higher diversity is thus inflated by augmented sentence length and is achieved for lower validity of the samples, as mentioned in the paragraph above.

We used BERT-base, DistilBERT-base, and

RoBERTa-base for fine-tuning. We used the versions of the models from Huggingface and found the best working hyperparameters via hyperparameter search. Hyperparameters with the QLoRA fine-tuning setup can be found in Appendix K. We trained each model 10 times per random seed and used 3 different random seeds with differently sampled seed samples and augmented samples for those seeds to avoid randomness of outcomes. The random seeds ensured that across various combinations, the same seeds and augmented samples could be used. The models were trained separately on the data collected from GPT-3.5 and Llama-3. As we aimed to compare the newer and established augmentation methods in a variety of cases, we used various numbers of seed samples per label and the number of collected samples per seed sample during fine-tuning. We ended up with a total number of fine-tunings (both full and using QLoRA) at 267,300, as we fine-tuned the models 10 times for each augmentation method, dataset, number of seed samples per label, number of collected samples per seed and random seed combination.

Finally, we computed the accuracy of all fine-tuned classifiers to allow their comparison.

4 Study Results

Our study has multiple parameter dimensions that together yield more than 11 thousand combinations. To keep the result presentation manageable, we collapse some of these dimensions (each of them with a different reasoning).

One dimension we could simplify are the augmentation methods themselves. To keep the comparison of *established* and *LLM-based* methods simple, we only compared the best-performing methods from each group (best downstream model accuracy). While the *established* method group contained 3 methods (given by the 3 augmentation techniques and their established implementations), the *LLM-based* method group contained 6 methods (the same 3 techniques, each implemented by 2 different LLMs[4]). We performed this comparison for each parameter combination of number of seeds, number of augmented samples per seed, classifier, fine-tuning approach, and dataset.

Among the LLM-based methods, the *paraphrasing* technique performed best in 56% cases, followed by *word insert*, which topped 30% of cases, and *word swap* with 14% of cases. Although *paraphrasing* performed best overall, the *word insert* worked best when the RoBERTa classifier was fine-tuned. Among the established methods, the *contextual word insert* performed best in 56% cases, followed by (backtranslation) *paraphrasing* topping 26% cases, and *contextual word swap* with 18% cases. Furthermore, the backtranslation had a stronger effect on classifier accuracy with full fine-tuning and lesser with QLoRA. Given these results, we decided to focus on the comparison of the LLM-based *paraphrasing* with the established *contextual word insert* methods. See appendix E for other method comparisons.

Another dimension we could collapse was the *number of collected samples per seed*, where we selected only the most accurate classifier for the same combination of other parameters. However, full details on how the number of collected samples per seed influences the classifier accuracy can be found in Appendix D.

4.1 Classifier Accuracy (RQ1)

To answer the RQ1, we compared the downstream classifier accuracy of LLM-based *paraphrasing* with the established *contextual word insert*, see Table 2. We counted the number of cases where one of these methods performed statistically significantly better than the other and also the number of cases where there was no statistically significant difference between the two methods, which we denote as the two methods having similar accuracy. For this, we used Mann-Whitney-U tests with *p=0.05*.

In most cases, the accuracy of LLM-based *paraphrasing* cannot be statistically distinguished from the *contextual insert*. However, when differences are observed, the LLM-based *paraphrasing* beats the *contextual insert* method in more cases. For full fine-tuning, this can be observed consistently (with the sole exception of RoBERTa with the News Category dataset). For QLoRA, the results are more mixed: while LLM-based *paraphrasing* generally yields better results for BERT and DistilBERT (with exceptions), for RoBERTa, the *contextual insert* surpasses the LLM-based *paraphrasing* more often. It should also be noted that of the three classifiers, RoBERTa performed best in ~80% of cases, as can be seen in Appendix L.

An investigation of the difference in mean ac-

[4] The results from Llama-3 and GPT-3.5 augmentation methods are both labeled as "LLM methods", as during the analysis, we found no significantly different model accuracy for augmentations created from the two LLMs used for training the classifier.

curacy between models trained using LLM *paraphrasing* and *contextual insert* can be found in Figure 2 for various number of seeds per label. Generally, a lower amount of seeds leads to a higher accuracy of classifiers trained on augmented data collected via the LLM *paraphrasing* than when *contextual insert* is used. This difference in accuracy is highest for 5 to 20 samples per label in cases where *paraphrasing* is more advantageous and decreases with more seed samples used.

There are also notable cases where *contextual insert* (a far cheaper augmentation method) provides better classifier accuracy than the LLM *paraphrasing*. This can be seen for BERT QLoRA fine-tuning in the FB dataset and Yelp dataset, and RoBERTa fine-tuning for the News Category dataset. BERT QLoRA exhibited results that favor one of the two methods more strongly than other types of fine-tuning and model combinations. In terms of increased classifier accuracy, when comparing fine-tuning with only seed samples themselves, both methods provide a relatively high increase of accuracy, compared to using only seed samples for training classifiers when using QLoRA (see visualization of this in Appendix G).

When considering the 3 fine-tuned models used, RoBERTa achieved the highest accuracy across all datasets. Considering this and the much more similar performance of the established and newer LLM-based methods for RoBERTa as seen in Figure 2, this could be indicating that even much cheaper established methods can achieve competitive model accuracy when compared to newer LLM-based augmentation methods on the best-performing classifier, with only exceptions for a small number of seeds per label.

We also did a combination of the *contextual insert* and *backtranslation* methods as the two best-established augmentation methods and compared it with the LLM-based methods, which did not result in a considerable increase in model precision compared to the *paraphrasing* method. Details of this comparison can be found in the Appendix E.

We answer *RQ1* as follows: in most cases, the established *contextual word insert* augmentation has a better or similar effect on classifier accuracy than the LLM-based *paraphrasing* augmentation. LLM methods perform better only with a small number of seeds per label. With an increasing number of seeds per label, the difference between the two methods for accuracy starts to diminish.

method	*Time cost*	*kgCO$_2$ emitted*	*Monetary cost*
Backtrans.	46m 40s	0.09	~$3
Con. swap	36m 40s	0.047	~$0.3
Con. insert	40m	0.047	~$0.3
Para. LLM	1h 10m	0.13	~$5
Swap LLM	1h 10m	0.13	~$5
Insert LLM	1h 10m	0.13	~$5

Table 1: Approximated kgCO$_2$ emitted, time, and monetary costs for each augmentation method on our hardware setup when collecting 15 samples for 100 seeds per label. The established methods take considerably less time and money while emitting far fewer emissions than newer LLM-based methods.

4.2 Analysis of Augmentation Costs and Benefits for Classifier Accuracy (RQ2)

To answer the RQ2, we first performed an approximate cost calculation for each of the used augmentation methods in terms of time needed to collect samples, monetary costs needed and emissions. We then identified cases where the higher cost of LLM-based methods is worth the increased accuracy of classifiers.

We measured the time needed to collect the given number of samples on the hardware that we used for the experiments. We measured the time needed for collecting 15 augmented samples per 100 seed samples per label. The newer LLM-based methods have the same estimated time needed for data augmentation for each of the methods, as we did not measure any significant differences in time needed between them. The results are displayed in Table 1.

Considering time only, the established augmentation methods run approximately 33%-47% faster than the newer LLM-based methods, and when considering also the CO$_2$ emissions, the *contextual swap* and *contextual insert methods* emit approximately 64% less kgCO$_2$ emissions for the same number of seed samples per label and number of collected samples. The details of how the emissions approximation calculation was done can be found in Appendix B. In terms of monetary costs, the *context swap* and *context insert* methods are approximately 16 times cheaper than the LLM-based methods. As such, the established text augmentation methods are considerably more efficient both in the time needed per collected sample, monetary cost, and in kgCO$_2$ emissions.

When considering the results in Figure 2, we observed increases in relative model classifier accuracy on small (5-20) number of seed samples when using the *paraphrasing* method compared to the

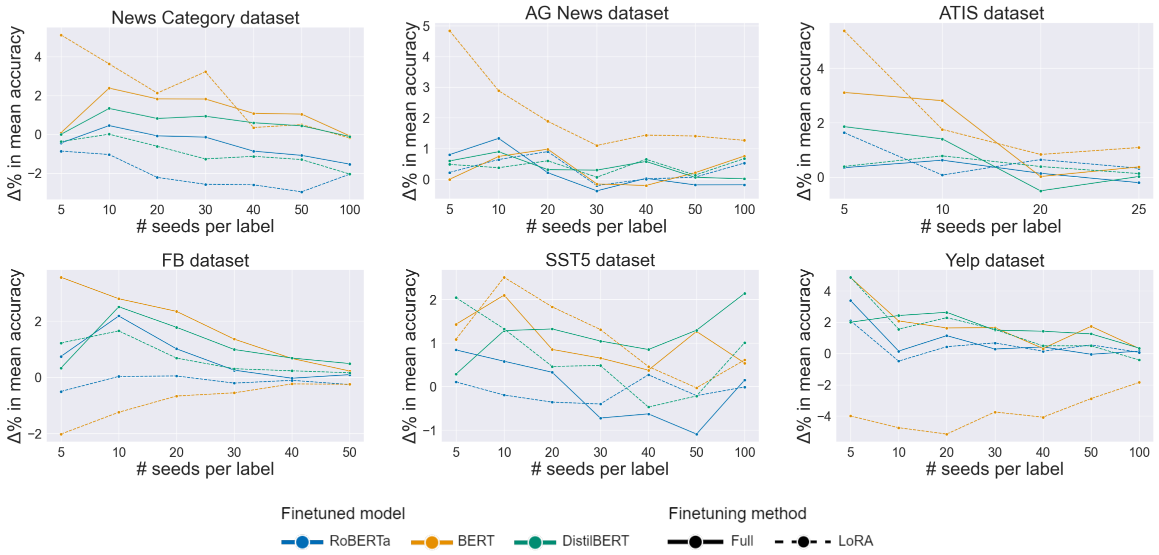

Figure 2: The difference in mean accuracy for classifiers trained on the *paraphrasing* augmentation method and the *contextual insert* augmentation method for 6 different datasets. The *paraphrasing* method generally works better for a small (5-20) number of seeds per label, and this benefit deteriorates with an increased number of seeds per label.

contextual insert method is 3%-17% better accuracy for classifiers when fine-tuning using QLoRA and 2%-11% when using full fine-tuning. However, for larger (30+) number of seed samples per label, the positive relative increase range decreases for QLoRA to 1.5%-6% and 0.5%-4% for full fine-tuning with a general increase of cases where the *contextual insert* method performed better for classifier accuracy. Although the differences in the relative increase in model performance when using *paraphrasing* method instead of *context insert* method decrease for a higher number of seeds, the difference in costs and emissions increases. This is most evident for RoBERTa, which had the smallest relative increase in accuracy out of all of the fine-tuned models, with some benefits only for a small number of seed samples per label used.

We answer *RQ2* as follows: Considering the results of increased classifier accuracy trained on the *paraphrasing* method augmentations for 5-20 seeds per label against those trained on the *contextual insert* method augmentations, the decreasing difference in accuracy between the methods with increasing number of seed samples per label and the augmentation methods cost approximation, the difference in accuracy seems to be worth the increased costs only for a small number of seeds. This is true for both full and QLoRA fine-tuning of models, while the difference in accuracy between the methods decreases significantly when using 30 seeds per label and more. Additionally, the cases where the *contextual insert* method is better for model accuracy increase with more seeds used.

5 Discussion

The results of our experiments lead to the following observations: First, the *paraphrasing* method was the best within the newer LLM-based augmentation methods, considering classifier accuracy. This could be due to the demonstrated ability (Cegin et al., 2023) of the newer LLMs to create very diverse paraphrases, being less constrained by seed samples. The *contextual insert* method worked best within the established augmentation methods. This may be caused by the *backtranslation* method creating a lot of duplicated samples and the *contextual swap* method introducing less variety than the *contextual insert* method.

Second, the number of cases in which the *paraphrasing* method (an LLM-based method) significantly outperforms the established *contextual insert* method decreases with more seed samples per label. This is similar to previous studies (Dai et al., 2023; Ubani et al., 2023), as we observed this in nearly all the cases for a small number of seeds and differs from the results of the previous study (Piedboeuf and Langlais, 2023), where such increase was observed less often. A different number of collected samples, classifiers used, and other factors might be the reason for this disparity. **The LLM-based methods achieve better classifier accuracy than established methods in the cases of very small**

Classifier→	RoBERTa						BERT						DistilBERT					
Dataset↓	Full			QLoRA			Full			QLoRA			Full			QLoRA		
AG News	7	68	1	13	50	4	14	54	1	24	36	0	7	70	0	17	44	3
News Category	2	58	11	1	12	35	19	44	1	19	38	4	15	52	1	3	18	30
ATIS	4	36	2	7	22	6	9	24	3	9	30	0	9	16	7	10	20	4
FB	20	28	2	5	42	10	25	22	0	3	28	19	25	20	1	20	16	8
SST-5	7	58	6	8	46	11	15	52	1	15	50	2	17	50	0	19	36	5
Yelp	11	56	3	15	42	6	19	42	2	0	8	38	17	48	1	31	12	5

Table 2: Comparison of the number of cases where models trained using data from either *paraphrasing* or *contextual insert* methods worked statistically *(p=0.05)* better or had similar accuracy when compared between each other. The numbers represent the result of one statistical test between 10 fine-tunings of the given model on data collected via the *paraphrasing* or the *contextual insert* using a specific random seed for a given number of seed samples per label. The cells are formatted in this way: `[# paraphrasing was better]` | `[# similar accuracy]` | `[# contextual insert was better]`. In most cases, the *paraphrasing* method works better for BERT and DistilBERT in both full fine-tuning and QLoRA fine-tuning.

seed numbers, which points to their potential benefits in low-resource settings.

Third, furthermore, when we increased the number of seeds, we observed a decrease of accuracy differences between models trained on data from *paraphrasing* and models trained with *contextual insert*, similar to (Piedboeuf and Langlais, 2023). The highest relative increase in model accuracy with *paraphrasing* instead of *contextual insert* appears with 5 to 20 seeds per label. After 30 or more seeds were used, the relative difference between methods decreased. Additionally, the difference between the LLM-based and established methods in terms of monetary costs, time costs, and emissions is quite significant (see section 4.2). **Therefore, it seems beneficial, from the perspective of both cost and model accuracy, to use the newer LLM-based augmentation methods only in low-resource settings**.

Fourth, we observed some exceptions to the trends reviewed above. **The fine-tuned RoBERTa models (which provided the best classification accuracy among the fine-tuned models) generally benefited more from augmentation methods that used insertion of words**. This might be due to a more robust pretraining of RoBERTa, where augmentations that introduce more noise are less beneficial for training. Another case was the fine-tuning of BERT using QLoRA, where, for some cases, the *paraphrasing* method was either considerably better or worse than the *contextual insert* method for classifier accuracy. This might be due to differences in the pre-training data and processes used for BERT in comparison with DistilBERT or RoBERTa, making it far more sensible to text augmentation methods when using QLoRA.

Fifth, the difference between the *paraphrasing* and the *contextual insert* method on model accuracy had much more variance for QLoRA than for full fine-tuning. When the *paraphrasing* method is used for QLoRA on classifiers, the increased accuracy (compared to *contextual insert*) is generally smaller than with full fine-tuning. **LLM paraphrasing's sample variability might be providing more benefits when the model can leverage it through full fine-tuning.**

Sixth, **the combination of the best-established methods does not improve their overall accuracy of the downstream model compared to using *contextual insert***. This might be due to the combination of methods leading to a possible distribution shift or models overfitting on the augmented data.

To summarize, as the costs of using established augmentation methods are considerably lower than the newer LLM-based methods and the increase in model accuracy decreases quickly with more seeds used, it appears to be beneficial to use them instead of newer LLM methods for a higher number of seeds per label when targeting model accuracy and use LLM-methods in cases of low-resource setting where the relative gain in accuracy is highest.

6 Conclusion

We compared the effects of newer *LLM-based* and *established* textual augmentation methods on downstream classifier accuracy for combinations of 6 datasets, 3 classifiers, 2 fine-tuning approaches, 2 augmenting LLMs, various numbers of seed samples per label and numbers of augmented samples per seed. In total, we analyzed a total of 267,300 fine-tunings. We aimed to identify cases where LLM-based augmentation outperforms established approaches in order to shed light on contrary results from previous studies. We identified the *paraphrasing* method as the best-performing LLM-based and the *contextual insert* as the best-performing established augmentation method. The comparison of these two best methods indicates that the use of LLM-based methods for data augmentation, instead of established methods, is only warranted for a small number of seed samples per label (5 to 20). There, we observed a statistically significant increase in cases where LLM-based methods are better and observed higher relative increases in model accuracy compared to established methods. However, with an increasing number of seeds per label, this effect decreased, and the number of cases of established methods having a higher influence on the accuracy of classifiers increased. As newer LLM methods are considerably more costly than established methods, their use is justified only for low-resource settings, where differences between the method's costs are smaller.

Limitations

We note several limitations to our work.

First, we only used datasets, augmentation methods, and LLMs for the English language and did not investigate cases of multi-lingual text augmentation.

Second, we did not use various patterns of prompts and followed those used in previous studies (Cegin et al., 2023; Larson et al., 2020). Different prompts could have effects on the quality of text augmentations, but they would also radically increase the size of this study, and thus, we decided to leave this for future work.

Third, we did not use newer LLMs for classification fine-tuning via PEFT methods (e.g., fine-tuning of Llama-3 or Mistral using QLoRA). While such inclusion would strengthen our findings, we decided not to use these models for classification fine-tuning due to two main reasons. First, the evaluation of these models is very costly and takes a long time due to their size, which results in them being mostly used with a small subset of the testing data (Chang and Jia, 2023; Li and Qiu, 2023; Gao et al., 2021; Köksal et al., 2023). This, in return, can lead to unintentionally cherry-picked results. Second, to do an analysis of this size for the combinations of parameters that influence one fine-tuning of models, we had to do a total of 44,500 fine-tuning for one model and fine-tuning method combination. Fine-tuning 44,500 times of a smaller generative LLM with 7B parameters and then evaluating it on a substantial split of the test data was infeasible to us time- and cost-wise. It would also radically increase the energy consumption of this study and, in turn, emissions emitted.

Fourth, from the family of PEFT methods we used only QLoRA and not multiple different PEFT methods. We opted for QLoRA due to its popularity and good performance. While including more fine-tuning methods in the paper would increase the strength of the findings and provide an even finer analysis of cases, it would also, similar to the case of not fine-tuning LLMs for classification from the previous limitation, lead to a significantly higher number of fine-tunings needed for a proper analysis of the new fine-tuning method added.

Fifth, for the LLM augmentation methods we used only Llama-3-8B and GPT-3.5. The results of data augmentation via LLama3 and GPT-3.5 yielded the same results on model accuracy. The inclusion of other LLMs in this type of study would considerably increase the number of fine-tunings of classifiers. It would likely provide no clear benefit for this study as previous works (Cegin et al., 2024) show little effect of various LLM on model performance. Additionally, we did not use larger models (e.g. 70B or GPT-4) as their increased performance in text augmentation for model accuracy has been shown (Cegin et al., 2024) to be not that significant when compared to variants of LLMs with fewer parameters.

Sixth, we only used 3 established methods compared to previous studies (Piedboeuf and Langlais, 2023; Dai et al., 2023; Ubani et al., 2023), which used more established methods for their comparisons. In our case, we used different types of methods, which had a good performance in previous studies (Dai et al., 2023; Piedboeuf and Langlais, 2023). While the inclusion of multiple other established methods would increase the strength of our findings, it would also require a lot of additional fine-tuning and evaluation to be done in order to get results for our detailed analysis.

Seventh, we did not enhance the LLM-based methods of *word insertion* or *word swap* with heuristics to select locations in seed texts where words should be replaced or added. We opted against this to let the LLMs decide internally (as a black box) which words to replace or add and where providing these methods with simplicity and without additional potential costs. A potential extension of these LLM methods with heuristics of where to replace or add words could possibly improve the performance of these methods for augmentation, and we see this as a natural extension of our work.

Eight, the *backtranslation* method could be improved by adding multiple languages into the translation process, which would possibly increase the lexical diversity of and number of created paraphrases. However, this would also increase the cost of using this method, which is already the most costly of all of the established augmentation methods.

Ninth, we only focus on classification tasks and make no claims about the effects of established and LLM-based text augmentation on other NLP tasks. However, as seen by the related work, classification constitutes an important task group, and even more so in low-resource settings.

Tenth, we do not know if any of the 6 datasets used in this study have been used for training the LLMs we used for data collection and if this had any effect on our results and findings. As such, we do not know how much would be the comparison of established and newer LLM augmentation methods different on new, unpublished datasets. This limitation is part of the recently recognized possible "LLM validation crisis", as described by (Li and Flanigan, 2023).

Acknowledgments

This work was partially supported by AI-CODE - AI services for COntinuous trust in emerging Digital Environments, a project funded by the European Union under the Horizon Europe, GA No. 101135437. https://doi.org/10.3030/101135437, DisAi - Improving scientific excellence and creativity in combating disinformation with artificial intelligence and language technologies, a project funded by European Union under the Horizon Europe, GA No. 101079164, https://doi.org/10.3030/101079164, and Modermed, a project funded by the Slovak Research and Development Agency, GA No. APVV-22-0414. The work was also partially funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the projects No. 09I03-03-V03-00020 and No. 09I01-03-V04-00006.

Part of the research results were obtained using the computational resources procured in the national project National competence centre for high performance computing (project code: 311070AKF2) funded by European Regional Development Fund, EU Structural Funds Informatization of society, Operational Program Integrated Infrastructure. This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254).

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7383–7390.

Jan Cegin, Branislav Pecher, Jakub Simko, Ivan Srba, Maria Bielikova, and Peter Brusilovsky. 2024. Effects of diversity incentives on sample diversity and downstream model performance in LLM-based text augmentation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13148–13171, Bangkok, Thailand. Association for Computational Linguistics.

Jan Cegin, Jakub Simko, and Peter Brusilovsky. 2023. ChatGPT to replace crowdsourcing of paraphrases for intent classification: Higher diversity and comparable model robustness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1889–1905, Singapore. Association for Computational Linguistics.

Ting-Yun Chang and Robin Jia. 2023. Data curation alone can stabilize in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8123–8144, Toronto, Canada. Association for Computational Linguistics.

Wenqing Chen, Jidong Tian, Liqiang Xiao, Hao He, and Yaohui Jin. 2020. A semantically consistent and syntactically variational encoder-decoder framework for paraphrase generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1186–1198, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jishnu Ray Chowdhury, Yong Zhuang, and Shuyi Wang. 2022. Novelty controlled paraphrase generation with retrieval augmented conditional prompt tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10535–10544.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang

Shen, Tianming Liu, and Xiang Li. 2023. Auggpt: Leveraging chatgpt for text data augmentation. *Preprint*, arXiv:2302.13007.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195, Toronto, Canada. Association for Computational Linguistics.

Luyang Fang, Gyeong-Geon Lee, and Xiaoming Zhai. 2023. Using gpt-4 to augment unbalanced data for automatic scoring. *Preprint*, arXiv:2310.18365.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Sreyan Ghosh, Chandra Kiran Evuru, Sonal Kumar, S Ramaneswaran, S Sakshi, Utkarsh Tyagi, and Dinesh Manocha. 2023. Dale: Generative data augmentation for low-resource legal nlp. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Sentosa, Singapore.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 238–252, Online. Association for Computational Linguistics.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990.*

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Abdullatif Köksal, Timo Schick, and Hinrich Schuetze. 2023. MEAL: Stable and active learning for few-shot prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 506–517, Singapore. Association for Computational Linguistics.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 737–762.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China. Association for Computational Linguistics.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700.*

Stefan Larson, Anthony Zheng, Anish Mahendran, Rishi Tekriwal, Adrian Cheung, Eric Guldan, Kevin Leach, and Jonathan K. Kummerfeld. 2020. Iterative feature mining for constraint-based data collection to increase data diversity and model robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8097–8106, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Changmao Li and Jeffrey Flanigan. 2023. Task contamination: Language models may not be few-shot anymore. *Preprint*, arXiv:2312.16337.

Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy Chen, Zhengyuan Liu, and Diyi Yang. 2023. CoAnnotating: Uncertainty-guided work allocation between human and large language models for data annotation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1487–1505, Singapore. Association for Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2024. Once: Boosting content-based recommendation with both open- and closed-source large language models. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 452–461, New York, NY, USA. Association for Computing Machinery.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Rishabh Misra. 2022. News category dataset. *arXiv preprint arXiv:2209.11429*.

Rishabh Misra and Jigyasa Grover. 2021. *Sculpting Data for ML: The first act of Machine Learning*. Independently published.

Aytuğ Onan. 2023. Srl-aco: A text augmentation framework based on semantic role labeling and ant colony optimization. *Journal of King Saud University - Computer and Information Sciences*, 35(7):101611.

Frédéric Piedboeuf and Philippe Langlais. 2023. Is ChatGPT the ultimate data augmentation algorithm? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15606–15615, Singapore. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online. Association for Computational Linguistics.

Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. Zeroshotdataaug: Generating and augmenting training data with chatgpt. *Preprint*, arXiv:2304.14334.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *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. Association for Computational Linguistics.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *Computational Science – ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part IV*, page 84–95, Berlin, Heidelberg. Springer-Verlag.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yue Zhou, Chenlu Guo, Xu Wang, Yi Chang, and Yuan Wu. 2024. A survey on data augmentation in large model era. *Preprint*, arXiv:2401.15422.

A Ethical considerations

Based on a thorough ethical assessment performed on the basis of intra-institutional ethical guidelines and checklists tailored to the use of data and algorithms, we see no ethical concerns pertaining directly to the conduct of this research. We also ethically assessed our paraphrase validity crowdsourcing process from Appendix C via our intra-institutional ethical guidelines and found no ethical concerns. In our study, we analyzed existing data or data generated using various LLMs. During our manual checking of the data, we also ensured that the data contained no personal or offensive data. Although the production of new data through LLMs bears several risks, such as the introduction of biases, the small size of the produced dataset sufficient for experimentation is, at the same time, insufficient for any major machine learning endeavors where such biases could be transferred.

We follow the license terms for all the models and datasets we used (such as the one required for the use of the Llama-3 model) – all models and datasets allow their use as part of the research.

B Details of CO2 Emission calculation and emissions related to experiments

For the estimations, we used the MachineLearning Impact calculator presented in (Lacoste et al., 2019). For estimations of GPU emissions, we used hardware of type A100 PCIe 40/80GB (TDP of 250W), and for estimation of CPU emissions, we used hardware of type Intel Xeon Gold 6148.

We conducted the data collection and fine-tuning on a custom private infrastructure with 16 core CPUs, 64 GB RAM, and 4xA100 GPUs. For the LLM-based augmentation methods and *backtranslation* method, we used the GPU to collect data, while for the *context insert* and *context swap* methods, we used CPUs only.

Data collection via Llama-3 was conducted using a private infrastructure, which has a carbon efficiency of 0.432 kg CO_2/kWh. A cumulative 20 GPU hours of computation was performed on hardware of type A100 PCIe 40/80GB (TDP of 250W) for data collection.

Model fine-tuning for all of the fine-tuned models using either full fine-tuning or QLoRA fine-tuning was conducted using a private infrastructure, which has a carbon efficiency of 0.432 kg CO_2/kWh. Approximately a cumulative of 1100 GPU hours of computation was performed on hardware of type A100 PCIe 40/80GB (TDP of 250W) for data collection.

Total emissions together are estimated to be 120.96 kgCO_2, of which 0 percent were directly offset. We tried to reduce the generated emissions by using 4-bit quantization for Llama-3 data collection and QLoRA training.

C Augmented samples validity: checking process and results

For the process of checking the validity of the created augmented samples, we used our very own web app developed for this process. The users, who were the authors that also developed the app, were shown the seed samples, their labels, and one particular sample to validate. The authors/users all gave consent to the data collection process and had knowledge of how the data would be used. The instructions were *"Please decide if the augmented sample has the same meaning as the seed sentence and if it adheres to the label of the seed sentence."* The user was then able to either mark the sample as valid or not, with an additional optional checkbox to label the samples as 'borderline case' for possible revisions. As the seed sentence changed only once in a while (we first showed all the paraphrases from one seed sentence), this significantly reduced the cognitive load on the annotator. The users/authors then discussed together the 'borderline cases' where the users were not sure about the validity of created paraphrases.

Before evaluating the validity of each augmentation method and the samples it produces, we filtered for malformed augmented samples, empty samples, or duplicated samples as per (Cegin et al., 2023). There were no such samples detected for the newer LLM-based methods. We detected around 0.05%-0.5% of all augmented samples to be duplicated for the *contextual word insertion* and *contextual word replacement*. The worst number of duplicates was detected for the *backtranslation* method, with 80% of all collected augmented samples to be duplicated. This still meant that we collected at least 2 to 3 augmented samples per seed, and as such, we did not eliminate this method from further evaluation. For fine-tuning cases using the *backtranslation* method where more number of collected samples per seed than 3 were needed, we used all of the available collected unique augmented samples. This high number of duplicates might be due to the translation model limitations with repeating patterns, as well as using only one intermediary language as per the original paper.

D Effects of number of collected augmented samples from augmentation methods on model accuracy

In this section, we compare the effects of a number of collected augmented samples per seed sample on model accuracy. We noticed that QLoRA fine-tuning benefited from more collected augmented samples per seed sample than full fine-tuning of classifiers for all of the methods. RoBERTa and DistilBERT full fine-tuning generally needed only a few (less than 5) augmented samples per seed sample to achieve the best classifier accuracy across different augmentation methods. This might be due to the more robust pretraining process in the case of RoBERTa and the distillation training of DistilBERT, where pretrained weights of the models benefit less from more added noise to the dataset via an increased number of collected augmented samples. Bert's full fine-tuning had a similar trend as RoBERTa and DistilBERT, with the exception of inserting words-based methods. This might indicate that while a lot of noise might degrade the accuracy of fine-tuned Bert (as is the case for the *paraphrasing* method), the augmented samples from word insertion add just enough noise for the model to benefit from it. Additionally, with an increased number of seed samples, we observed that fewer augmented samples per seed sample were needed for the fine-tuned models to achieve the best accuracy, indicating that a lot of augmented examples in the training data could lead to a distribution shift. Visualization of these results can be found in Figure 3.

E Combination of best established augmentation methods for classifier accuracy

Given that the established augmentation methods are cheaper when compared to newer LLM-based augmentation methods, we can combine the established augmentation methods together and then compare them with newer LLM-based methods, specifically the *paraphrasing method*, to determine if such combination increases the accuracy of models for classification.

To do so, we combine the *backtranslation* and *contextual insert* method in this way: for each number of collected samples per seed sample, we use from both methods augmented samples, e.g. for 2 number of collected samples used per seed sample from the *paraphrasing* method we include 2 from the *backtranslation* method and 2 from the *contextual insert* method. As mentioned in Appendix C, the *backtranslation* method produces a lot of duplicate samples, meaning that for cases where not enough unique augmented samples are collected, we used all of the available augmented samples.

The comparison of the combination of the established methods and the *paraphrasing* method for classifier accuracy can be seen in Table 3. Compared to the results from Section 4.1 where we compared the *paraphrasing* method against only the *contextual insert* method, the combination of the two best-established methods yields more cases where the *paraphrasing* method is better for all the fine-tuning methods and dataset combinations. This might be due to such a combination of two augmentation methods introducing a potential distribution shift in the data, overfitting on augmented data, or possible inconsistencies in the augmented data. To conclude, the combination of the established methods increases the cost of augmentation while providing worse results compared to only using the *contextual insert* method for classifier accuracy.

F Results for other combinations of LLM-based and established methods on model accuracy

We also compared other methods with the best-performing LLM-based method (*paraphrasing*) and the best-established method (*contextual insert* method). When comparing the *paraphrasing* method and *contextual swap* method, we can see based on Table 5 and Figure 5 that the *paraphrasing method* is in nearly all cases better than the *contextual swap* method, but the increased model accuracy decreases with number of seeds per label. A similar comparison can be seen for the *paraphrasing* method and the *backtranslation* method in Table 4 and Figure 4.

When comparing the best-established method *contextual insert* with the *insert word* LLM-based method, we can see that it performs better for RoBERTa finetuning in Table 6, but the increase in model accuracy is not high as seen in Figure 6. When comparing the *contextual insert* method and the *swap word* LLM-based method, the difference is even more in favor of the *contextual insert* method as seen in Table 7 and Figure 7.

In general, the swap word methods performed the worst, while the insert words methods performed the best in cases of finetuning robustly pretrained models (RoBERTa) or for noisy datasets (SST-5, Yelp).

G Comparison of augmentation methods increase for models accuracy against training only with seed samples

We compared the best LLM-based augmentation method *paraphrasing* and the best-established augmentation method *contextual insert* and their effects on model accuracy when compared to models trained only using the seed samples. The results can be seen in Figures 8 and 9. The LoRA fine-tuning methods have the highest relative and absolute increase for model accuracy, even when considering increasing the number of seed samples per label. Even though this increased accuracy decreases with the number of seed samples used, this is most prominent for full fine-tuning, where cases of negative difference of mean accuracy exist. For LoRA finetuning, the increased accuracy is still

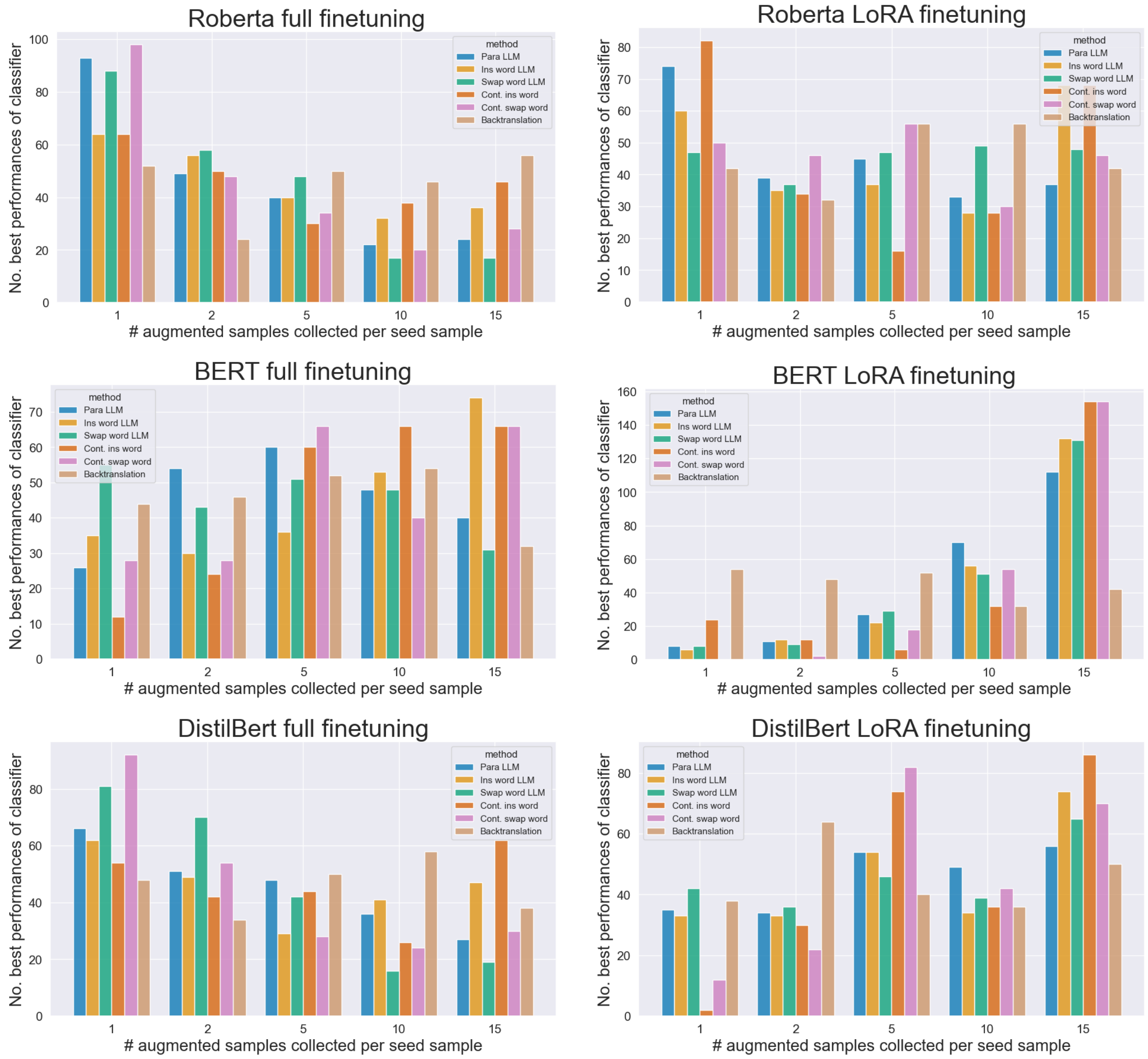

Figure 3: The number of cases per number of collected augmented samples per seed sample where each augmentation method achieved the best accuracy for 6 different combinations of models and fine-tuning methods. Except for RoBERTa and DistilBERT full fine-tuning, the methods worked best for model accuracy when more augmented samples were provided.

CLASSIFIER→	ROBERTA						BERT						DISTILBERT					
Dataset↓	Full			QLoRA			Full			QLoRA			Full			QLoRA		
AG News	7	70	0	16	46	3	12	54	3	30	24	0	8	62	3	25	30	2
News Category	3	60	9	1	12	35	19	44	1	30	24	0	10	62	1	3	26	26
ATIS	12	18	3	14	12	4	10	26	1	12	22	1	12	16	4	12	22	1
FB	11	42	4	11	32	9	21	22	4	36	0	0	16	32	4	19	14	10
SST-5	5	72	1	5	48	13	9	64	1	21	42	0	14	54	1	20	26	9
YELP	7	58	6	9	46	10	16	52	0	40	4	0	11	58	2	26	20	6

Table 3: Comparison of the number of cases where models trained using data from either *paraphrasing* or a combination of *contextual insert* and *backtranslation* methods worked statistically (p=0.05) better or had similar accuracy when compared between each other. The numbers represent the result of one statistical test between 10 fine-tunings of the given model on data collected via the *paraphrasing* or the combination of two methods using a specific random seed for a given number of seed samples per label. The cells are formatted in this way: `[# paraphrasing was better]` | `[# similar accuracy]` | `[# combination was better]`. In most cases, the *paraphrasing* method works better for BERT and DistilBERT in both full fine-tuning and QLoRA fine-tuning, with a decrease of such cases when fine-tuning RoBERTa.

CLASSIFIER→	RoBERTa						BERT						DistilBERT					
Dataset↓	Full			QLoRA			Full			QLoRA			Full			QLoRA		
AG News	6	72	0	17	44	3	19	46	0	42	0	0	8	64	2	37	10	0
News Category	4	58	9	6	4	34	27	30	0	42	0	0	7	60	5	33	6	6
ATIS	14	16	2	16	10	3	16	16	0	20	8	0	13	20	1	20	6	1
FB	24	22	1	36	0	0	32	8	0	36	0	0	28	16	0	36	0	0
SST-5	11	52	5	37	8	1	22	38	1	41	2	0	4	76	0	42	0	0
YELP	4	52	12	16	32	10	22	36	2	6	6	33	10	62	1	37	2	4

Table 4: Comparison of the number of cases where models trained using data from either *paraphrasing* or *backtranslation* method worked statistically (p=0.05) better or had similar accuracy when compared between each other. The numbers represent the result of one statistical test between 10 fine-tunings of the given model on data collected via the *paraphrasing* or the *backtranslation* using a specific random seed for a given number of seed samples per label. The cells are formatted in this way: `[# paraphrasing was better]` | `[# similar accuracy]` | `[# backtranslation was better]`. In nearly all cases the *paraphrasing* method works better for model accuracy, except for the Yelp dataset, where smaller changes from the *backtranslation* might be more beneficial.

CLASSIFIER→	RoBERTa						BERT						DistilBERT					
Dataset↓	Full			QLoRA			Full			QLoRA			Full			QLoRA		
AG News	4	70	3	13	42	8	5	54	10	22	38	1	9	52	7	17	36	7
News Category	11	52	5	5	8	33	32	18	1	33	14	2	29	26	0	5	22	26
ATIS	15	14	2	15	14	2	22	4	0	20	8	0	18	12	0	20	6	1
FB	36	0	0	36	0	0	35	2	0	33	6	0	36	0	0	36	0	0
SST-5	19	44	1	22	26	7	29	26	0	31	22	0	32	20	0	39	6	0
YELP	23	38	0	26	26	3	33	18	0	40	4	0	26	30	1	37	10	0

Table 5: Comparison of the number of cases where models trained using data from either *paraphrasing* or *contextual swap* method worked statistically (p=0.05) better or had similar accuracy when compared between each other. The numbers represent the result of one statistical test between 10 fine-tunings of the given model on data collected via the *paraphrasing* or the *contextual swap* using a specific random seed for a given number of seed samples per label. The cells are formatted in this way: `[# paraphrasing was better]` | `[# similar accuracy]` | `[# contextual swap was better]`. In most cases, the *paraphrasing* method works better for model accuracy in nearly all cases, with a higher accuracy of the *contextual swap* method on news classification datasets.

CLASSIFIER→	RoBERTa						BERT						DistilBERT					
Dataset↓	Full			QLoRA			Full			QLoRA			Full			QLoRA		
AG News	5	74	0	13	54	2	5	72	1	21	40	1	6	68	2	16	36	8
News Category	8	52	8	1	12	35	4	58	9	3	54	12	6	64	4	1	6	38
ATIS	9	28	1	10	20	4	2	38	3	4	24	8	4	34	3	3	2	20
FB	18	34	1	20	18	7	12	36	6	12	6	21	19	30	2	26	6	7
SST-5	4	74	1	6	54	9	3	74	2	7	64	3	2	70	5	15	38	8
YELP	4	74	1	5	64	5	6	62	5	0	4	40	0	76	4	5	30	22

Table 6: Comparison of the number of cases where models trained using data from either *insert words* LLM-based method or *contextual insert* method worked statistically (p=0.05) better or had similar accuracy when compared between each other. The numbers represent the result of one statistical test between 10 fine-tunings of the given model on data collected via the *insert words* or the *contextual insert* using a specific random seed for a given number of seed samples per label. The cells are formatted in this way: `[# insert words was better]` | `[# similar accuracy]` | `[# contextual insert was better]`. The *swap word* method works well for RoBERTa full fine-tuning, but other than that, the cases where it outperforms the *contextual insert* method are equal to the cases where it is outperformed.

relatively high, no matter the number of seeds per label.

H Dataset details

As we did not use all of the dataset labels and samples in each of the datasets, we list our setup here. We mostly used labels that were in the datasets with similar quantities to deal with the imbalanced datasets issue. All used datasets are in English language. For the *News Category* dataset, we used samples with labels *politics*, *wellness*, *entertainment*, *travel*, *style and*

CLASSIFIER→	RoBERTa						BERT						DistilBERT					
Dataset↓	Full			QLoRA			Full			QLoRA			Full			QLoRA		
AG News	1	80	1	9	52	7	4	76	0	0	70	7	7	68	1	17	40	5
News Category	1	42	20	0	8	38	0	46	19	0	38	23	0	54	15	0	2	41
ATIS	0	20	14	3	14	14	0	12	18	0	14	17	1	14	16	0	2	23
FB	0	18	27	0	4	34	0	12	30	0	4	34	0	20	26	1	0	35
SST-5	3	70	4	9	52	7	2	78	1	3	72	3	5	60	7	15	32	11
YELP	0	68	8	2	38	21	1	54	14	4	30	23	0	54	15	4	14	31

Table 7: Comparison of the number of cases where models trained using data from either *swap words* LLM-based method or *contextual insert* method worked statistically (p=0.05) better or had similar accuracy when compared between each other. The numbers represent the result of one statistical test between 10 fine-tunings of the given model on data collected via the *swap words* or the *contextual insert* using a specific random seed for a given number of seed samples per label. The cells are formatted in this way: `[# swap words was better]` | `[# similar accuracy]` | `[# contextual insert was better]`. The *swap words* method is generally worse than the *contextual insert* method for model accuracy.

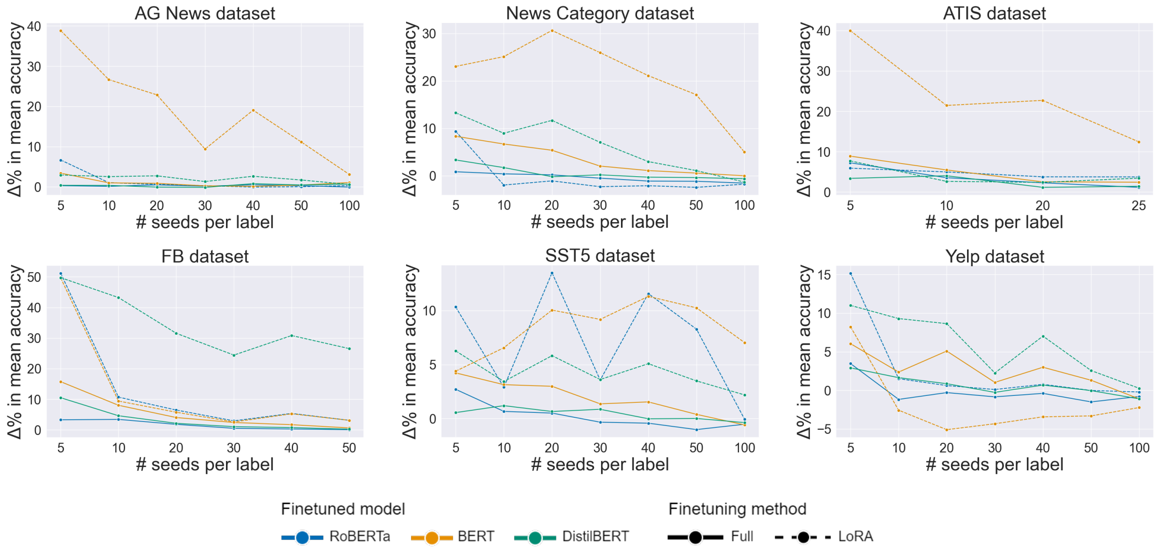

Figure 4: The difference in mean accuracy for classifiers trained on the *paraphrasing* augmentation method and the *backtranslation* augmentation method for 6 different datasets. The *paraphrasing* method works generally better in all cases.

beauty, and *parenting*. For the *AG News*, *SST-5*, and *Yelp* datasets, we used all the samples. For the *ATIS* dataset we used samples with labels *atis_abbreviation*, *atis_aircraft*, *atis_airfare* and *atis_flight_time*. For the *FB* dataset we used samples with labels *get_directions*, *get_distance*, *get_estimated_arrival*, *get_estimated_departure*, *get_estimated_duration*, *get_info_road_condition* and *get_info_traffic*. For the ATIS dataset, we used values for a number of seed samples per label [5, 10, 20, 25], and for the FB dataset, we used values [5, 10, 20, 30, 40, 50] as both of these datasets had classes with fewer number of samples.

I Established augmentation methods parameters used

For the *backtransaltion* method, we used the *facebook/wmt19-de-en* and *facebook/wmt19-en-de models* models and set the maximum length of the produced translations to 300.

For the *contextual insert* and *contextual swap* methods same parameters were used: we considered 100 tokens for augmentation, with 30% of the input text being changed with a minimum of 1 word and maximum of 10 words being either swapped or added and used *BERT-large-uncased* [5] for our experiments.

[5] https://huggingface.co/google-bert/bert-base-uncased

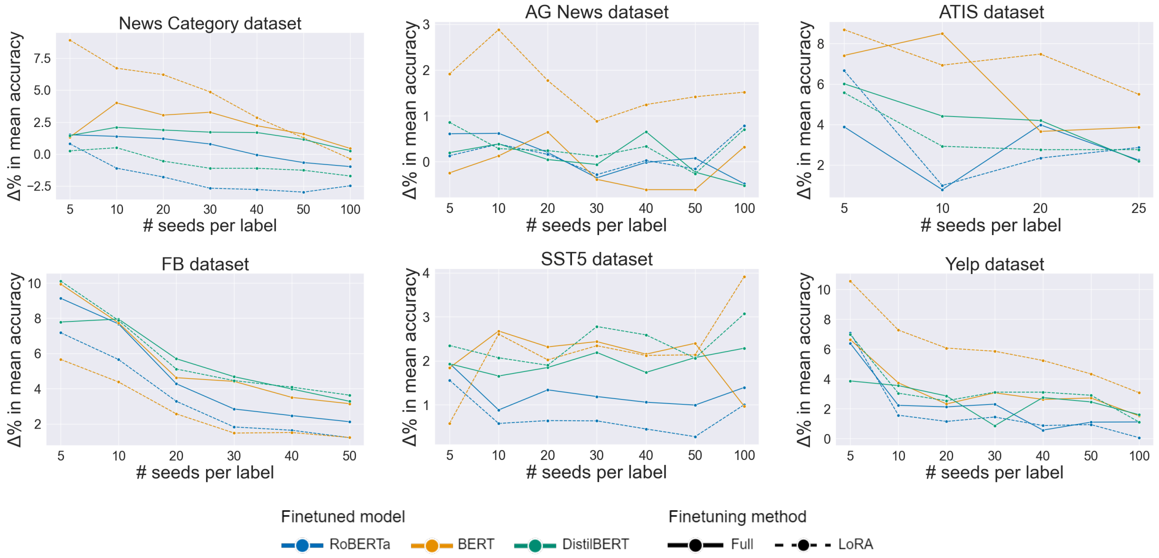

Figure 5: The difference in mean accuracy for classifiers trained on the *paraphrasing* augmentation method and the *contextual swap* augmentation method for 6 different datasets. The *paraphrasing* method works generally better in all cases with a decreasing effect with an increased number of seeds per label.

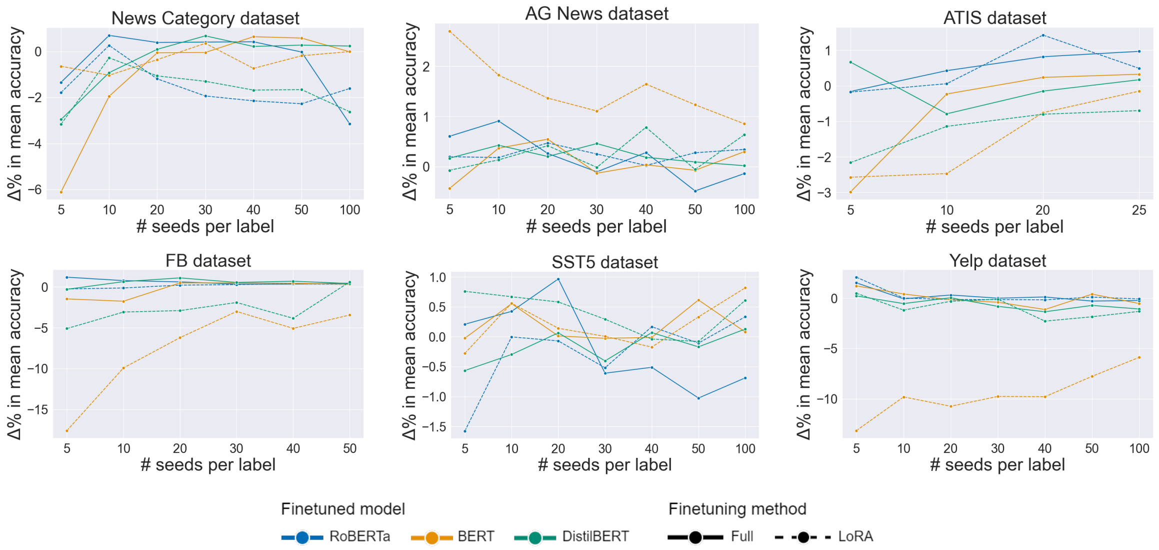

Figure 6: The difference in mean accuracy for classifiers trained on the *insert words* LLM-based augmentation method and the *contextual insert* augmentation method for 6 different datasets. The cost of using the *insert words* LLM-based method outweighs the benefits, as the *contextual insert* method works in many cases slightly worse or outright better for model accuracy.

J LLM-based augmentation methods parameters and templates used

For GPT-3.5 data collection, we used the *gpt-3.5-turbo-0125* version of the model with *temperature* of 1, *top p* of 1 and *presence penalty* at 0. For Llama3-8B, we used the *instruct* version [6], 4-bit quantization, max new tokens set at 1024, *temperature* of 0.1 and *top p* of 1. We collected 1 response for each seed sentence as we asked for 15 different augmentations in our prompts, which are listed below. Both LLMs used the same prompts.

Paraphrasing prompt: *Please provide 15 different changes of the Text by paraphrasing it. Output the full sentences. Output in format "1. sentence 1, 2. sentence 2, ..., 15. sentence 15". Text: "seed text placeholder".*

Insert words prompt: *Please provide 15 different*

[6] https://huggingface.co/meta-Llama/Meta-Llama-3-8B-Instruct

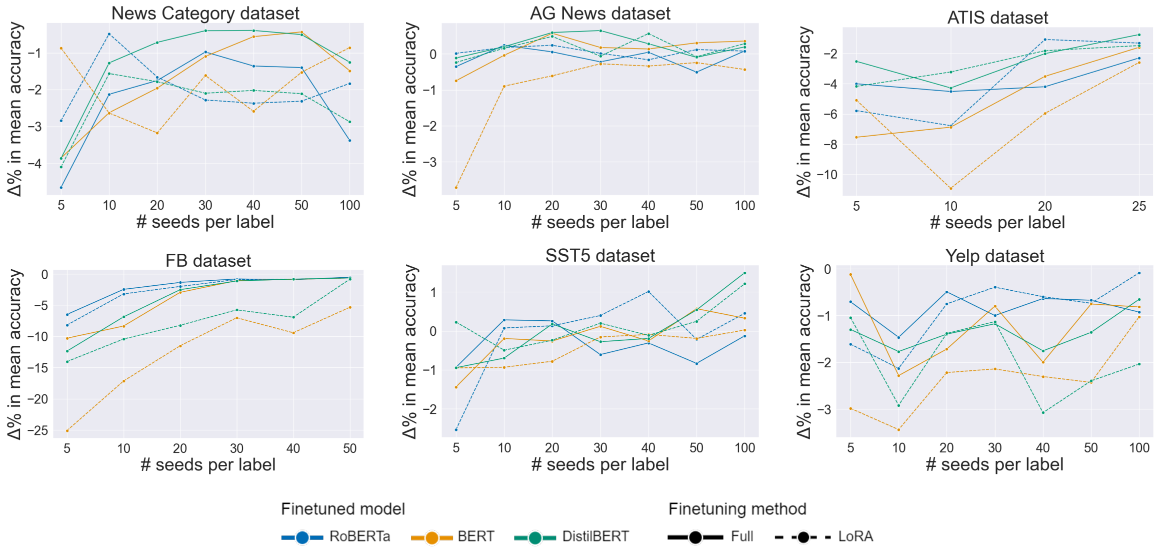

Figure 7: The difference in mean accuracy for classifiers trained on the *swap words* LLM-based augmentation method and the *contextual insert* augmentation method for 6 different datasets. The cost of using the *swap words* LLM-based method outweighs the benefits, as the *contextual insert* method works in many cases better for model accuracy.

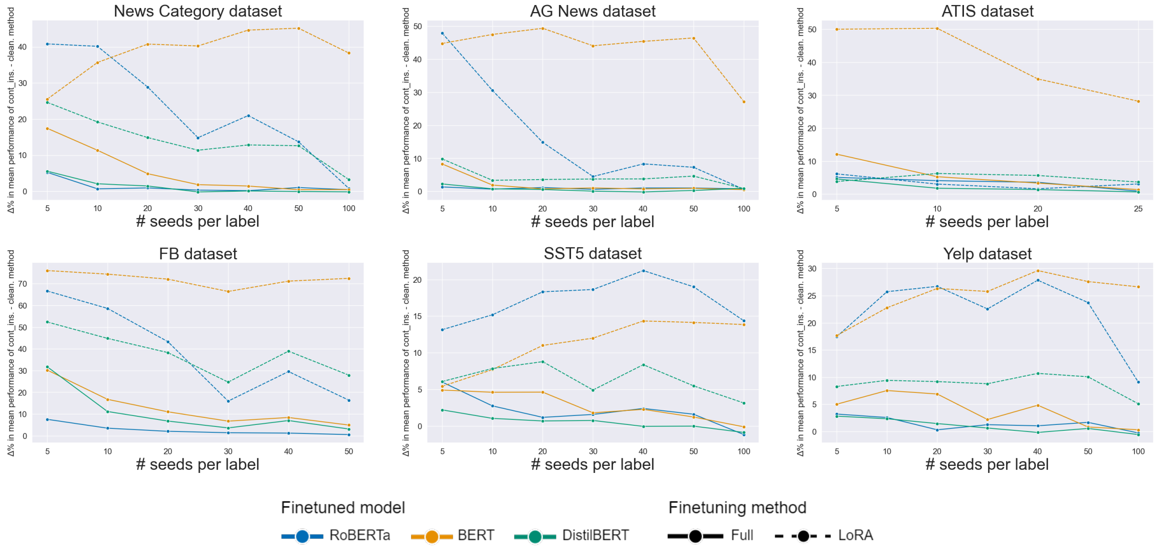

Figure 8: The difference in mean accuracy for classifiers trained on the *contextual insert* established augmentation method and using only the seed samples for fine-tuning for 6 different datasets. The cost of using the *swap words* LLM-based method outweighs the benefits, as the *accuracy insert* method works in many cases better for model performance.

changes of the Text by inserting words into the Text. Output the full sentences. Output in format "1. sentence 1, 2. sentence 2, ..., 15. sentence 15". Text: "seed text placeholder".

Swap words prompt: *'Please provide 15 different changes of the Text by swapping words for their synonyms. Output the full sentences. Output in format "1. sentence 1, 2. sentence 2, ..., 15. sentence 15". Text: "seed text placeholder".*

K Classifier fine-tuning details

We selected the best hyperparameters after using a hyperparameter search across models and classifiers. For both full-finetuning and LoRA finetuning, we used the same batch size across classifiers based on the number of seed samples per label: we used 16 batch sizes for 5 to 20 seeds per label, 32 batch sizes for 20 to 30 seeds per label and 64 for 40 and more seeds per label. We used the same learning

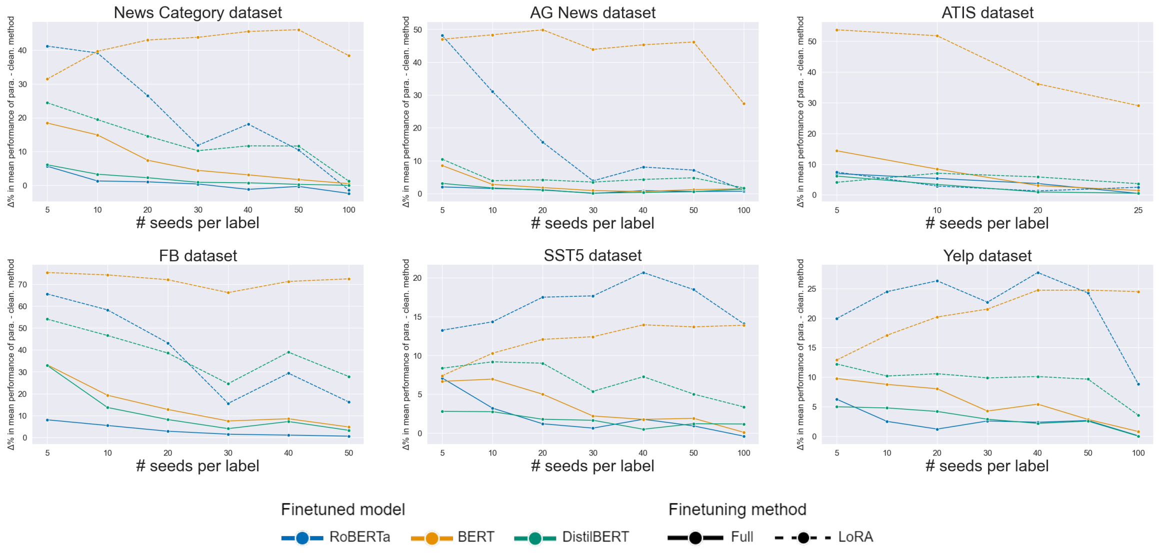

Figure 9: The difference in mean accuracy for classifiers trained on the *paraphrasing* LLM-based augmentation method and using only seed samples for fine-tuning for 6 different datasets. The cost of using the *swap words* LLM-based method outweighs the benefits, as the *contextual insert* method works in many cases better for model accuracy.

rate across classifiers set at *1e-4*. We used AdamW optimizer in all cases.

For LoRA finetuning, we used *r=16*, *alpha=16*, *dropout=0.1* and trained the model for 80 epochs. For full-finetuning, we performed the fine-tuning for 30 epochs.

L Best classifier model results

We investigated which classifier performed best for both full fine-tuning and LoRA fine-tuning. We performed this analysis when comparing the *paraphrasing* LLM-based method and *contextual insert* method. We compared the cases with the same dataset, number of seed samples per label, and random seeds used. In the majority of cases (approximately 80% of the time), fine-tuned RoBERTa had the highest accuracy in all cases of fine-tuning, followed by DistilBERT and then BERT. The visualization of the results can be seen in Figure 10.

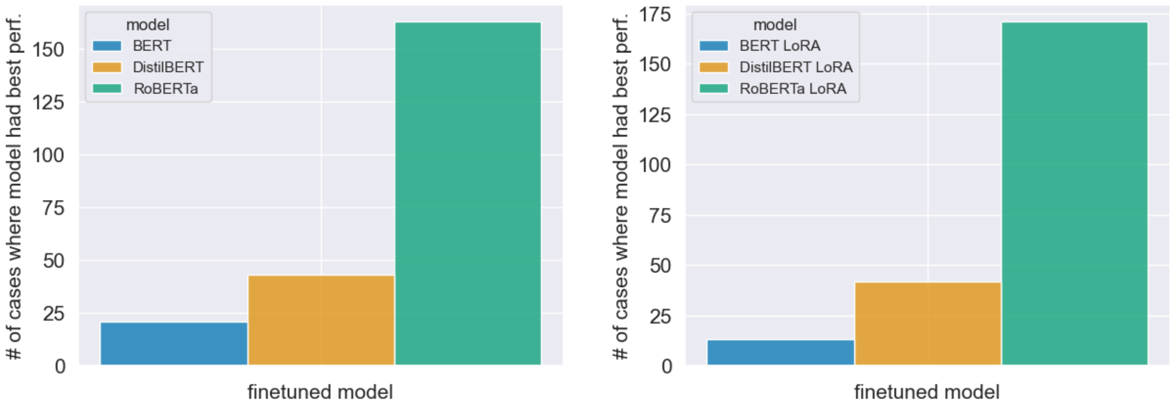

Figure 10: No. cases where each model achieved the highest accuracy for a particular combination of number of seeds, collected seeds, and dataset when using full fine-tuning (left) and LoRA fine-tuning (right). These cases were gathered from the comparison of *paraphrasing* LLM-based augmentation method and *contextual insert* augmentation method.