Can you give me a boost? - Increasing Performance of Fine-Tuned Language Model Through Paraphrasing

An investigation of using LLM-generated paraphrases to boost the fine-tuning performance of language models in low-resource domains or languages.

Jørgen Højlund Wibe (201807750)
Niels Krogsgaard (202008114)

Supervisor: Roberta Rocca
Natural Language Processing · Cognitive Science
School of Communication and Culture, University of Aarhus
January 3th 2024

Paper *GitHub* Repository:
`https://github.com/jorgenhw/NLP_exam_2023`

Characters in total: 52,117

Abstract

Advancements in natural language processing (NLP) have brought substantial improvements to language models, particularly with the rise of transformer architecture and Large Language Models (LLMs). Despite their prowess, a significant challenge remains for low-resource languages like Danish, where digital and annotated data are scarce, and NLP tools are limited. This paper explores the potential of using multilingual LLMs to enhance Danish text data through data augmentation (DA), specifically by paraphrasing with an LLM to enrich training datasets for improved model performance.

We use Mistral 7B Instruct, an open-source multilingual LLM, for paraphrase generation. Through prompt engineering, we guide the LLM to generate Danish paraphrases of a sentiment classification dataset originating from Twitter. A multistage filtering process is applied to ensure the validity of data labels while maximising the diversity. The augmented dataset is then evaluated by fine-tuning a BERT-Large model to different versions of the dataset.

Evaluation shows that the generated paraphrases does not improve performance, but rather the performance stays the exact same. A post-hoc investigation showed that the results did not change if GPT-4 was used for paraphrase generation, or if the dataset was changed. Reasons for this lack of an effect are discussed with the authors pointing to the paraphrasing not generating enough diversity or lack of capabilities of the BERT model.

Keywords: *NLP, machine learning, data augmentation, paraphrasing*

Contents

1 Introduction

The past few years have seen a massive increase in the development and capabilities of language models, partially ignited by the introduction of the transformer model architecture in 2017 and fueled by the finding that bigger is better, resulting in what is referred to as Large Language Models (LLMs) (Vaswani et al., 2017; Kaplan et al., 2020a). While the advancements have been remarkable, one area that presents significant challenges is their performance with low-resource languages for which there is limited digital data available and less research done in Natural Language Processing (NLP) (Bender, Emily, 2019; Ruder, Sebastian, 2019). This has sprouted the growth of a new field within NLP research that tries to solve the problem of data scarcity for 'small' languages. Methods such as transfer learning, multilingual models, distant supervision, and data augmentation (DA) are all investigated and proposed as parts of a possible solution. This will not only make NLP tools more available in low-resource languages but also help data-scarce domains in high-resource languages (Hedderich, Lange, Adel, Strötgen, & Klakow, 2021).

This paper will investigate the possibility of using a multilingual LLM to perform data augmentation, generating paraphrases on a dataset from a low-resource language, i.e. Danish. The aim is to improve the performance of small language-specific language models by augmenting the training data used to fine-tune the model.

1.1 Low resource languages

NLP faces a significant challenge in addressing the extreme diversity of the world's languages. In NLP, a distinction is often made between high- and low-resource languages. In reality, it is more like a continuum. The distinction comes from the fact that the vast majority of NLP research is focused on around 10-20 out of the world's languages, leaving the remainder underrepresented in the field (Magueresse, Carles, & Heetderks, 2020; Hedderich et al., 2021). This has led to a deficit in several crucial elements for the development of sophisticated NLP tools in these less-represented languages, including the availability of high-quality data, annotated corpora, computational resources, academic research, and commercial interest (Magueresse et al., 2020). One example is the training data for GPT-3. Johnson et al. (2022) specifically look at the proportion of training data going into training OpenAI's GPT-3 compared to the proportion of speakers of a language (Table 1) which highlights the distortion in language representation (Johnson et al., 2022).

Whether something is a low resource language depends a lot on the specific task, since some tasks can be solved well with a low amount of data, while others seem to only be solved well with a large amount of data (Hedderich et al., 2021). Additionally, a study by Plank, Søgaard, and Goldberg (2016) found that different languages also have different data and model requirements. They found that the performance of a model varied across languages given the same amount of training data (Plank et al., 2016). In the work by X. Zhang et al. (2023) on

GPT-3 training data (2019)	English (93%)	French (1.8%)	German (1.5%)	Spanish (0.8%)	Italian (0.6%)
First-language spoken (2019)	Mandarin (12%)	Spanish (6%)	English (5%)	Hindi (4.4%)	Bengali (4%)
Most spoken language (2021)	English (1348M)	Mandarin Chinese (1120M)	Hindi (600M)	Spanish (543M)	Standard Arabic (274M)

Table 1: The proportion of training data going into GPT-3 versus the proportion of language speakers in the world. Source: (Johnson et al., 2022, p. 3)

Twitter text data, Danish is categorised as a low-resource language due to low representation in their dataset compared to the other relevant languages.

1.2 Language modelling

Being able to even suggest paraphrasing text using language models, is only possible due to the last decade's advancements in NLP. This section outlines the newer developments in NLP and comments on the so-called scaling law.

A pivotal turning point in the development of language models was marked by the introduction of the transformer architecture by Vaswani et al. (2017). Preceding this breakthrough, the field relied on convolutional neural networks, recurrent neural networks, and various iterations of Long Short-Term Memory (LSTMs) models which often struggled to effectively integrate contextual information during text processing (Khurana, Koli, Khatter, & Singh, 2023). The development of the self-attention mechanism of the transformer architecture addressed this issue. It processes input sequences through three distinct matrix transformations learned during pre-training — queries, keys, and values—to create representations that highlight the importance of specific words in relation to others. Demonstrated as a highly efficient approach, it does not use recurrence as in RNN, and through having multi-head attention different subspaces of the input sequence can be modelled and then combined (Vaswani et al., 2017). The original proposed transformer was an encoder-decoder architecture but has since developed into three overall categories of transformers:

1. the *encoder-only* transformers, which can be referred to as the BERT-style architectures, since the BERT model has been a foundational and successful implementation of this (Devlin, Chang, Lee, & Toutanova, 2019)
2. the *decoder-only* transformers, which the GPT architecture is an example of (Radford, Narasimhan, Salimans, & Sutskever, 2018)[1]
3. the *encoder-decoder* transformers, which the T5 architecture is an example of (Raffel et al., 2020)

[1] For the remainder of this paper, we refer to the decoder-only models as LLMs and encoder-only as BERT models or language models.

The superior performance of the transformer architecture has generated a massive amount of development and research with new and better models being released almost by the month. For some time, a general trend for training all the different transformer models has been following a simple scaling law: bigger is better. Back in 2020, Kaplan et al. (2020b) investigated this scaling hypothesis and saw that training loss decreases according to a power law with increasing training time, dataset size, and number of model parameters. This has caused a focus on training and releasing models that are rapidly increasing in size to create better performance as seen in Figure 1.

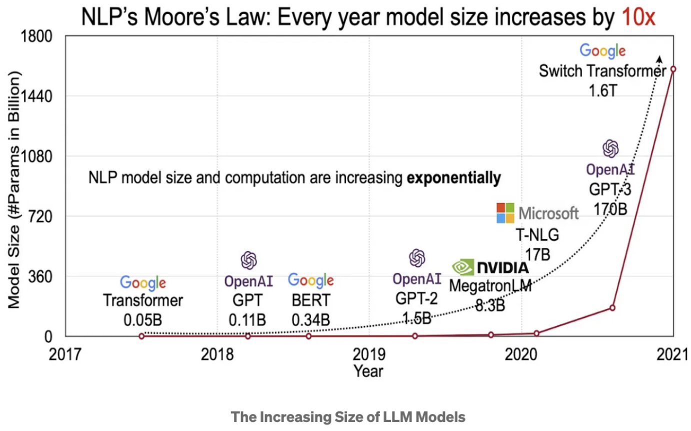

Figure 1: Figure showing the size of different large language models over time (Harishdatalab, 2023)

However, research has found that the scaling law does not always hold (McKenzie et al., 2023). Smaller and well-performing models are released that often have comparable or better performance than the biggest models (Hoffmann et al., 2022; Touvron, Lavril, et al., 2023). Model sizes are decreased through more clever training (Hoffmann et al., 2022), improvement of data quality (Gunasekar et al., 2023), and knowledge distillation from bigger models (Hsieh et al., 2023). This makes inference from models more efficient and makes the models accessible to more people and institutions since they can be run on a single or few GPUs instead of needing a complex and expensive computing architecture (Hsieh et al., 2023).

Moreover, research has found that encoder-only models (BERT) in some cases outperform decoder-only models (LLMs). Hsieh et al. (2023) shows that task-specific small models beat the best-performing general-purpose LLMs. In a comparison by Wang, Yu, Firat, and Cao (2021) they show that GPT-3 with a few-shot prompt is on par with a basic fine-tuned BERT model, while more advanced BERT architectures and T5 are performing on average 18 percentage points better than GPT-3 (Wang et al., 2021). The relatively small size of pre-trained BERT-style models makes them easier and less computationally extensive to train, fine-tune, and deploy. This also makes them much more available in many variants, including language-specific types with good performance in low-resource languages.

Taking these insights into account, the subsequent section dedicates a focused examination of the fine-tuning process itself as this process is also relevant to this paper.

1.3 Fine-tuning and prompt-engineering

This section explains the idea behind fine-tuning and to what extent the amount of data affects this, followed by a short introduction to prompt engineering.

Fine-tuning generally refers to the process of adapting a pre-trained model to a more specific downstream task (Devlin et al., 2019). The exact process of fine-tuning depends on the type of pre-trained model and the task. In this paper, we refer to fine-tuning as the process of training a BERT-style model on labelled task-related data. When fine-tuning a BERT model, an additional layer is added to the model before training on the specific task (Devlin et al., 2019). For a text classification task, the added layer is used to map the learned representations to a probability distribution of the specific number of classes. Training the model adapts its parameters to solve the given task through supervised learning (Devlin et al., 2019). The final performance of the fine-tuned model depends, among other things, on the capabilities of the language model, difficulty of the task, quality of the fine-tuning data, and amount of fine-tuning data (Devlin et al., 2019; Mosbach, Andriushchenko, & Klakow, 2021; Dodge et al., 2020a). These elements interact in many unpredictable ways, but if one of them is improved it will most likely affect the performance of the resulting model.

1.3.1 The effect of data amount on fine-tuning performance

Directly relevant to this paper is how the amount of data used in fine-tuning affects the performance of the model.

In a study by Mosbach et al. (2021) they found that increasing the amount of training data increased the generalization of the fine-tuned language model, especially when the initial dataset was relatively small. Another NLP study by Leite, Silva, Bontcheva, and Scarton (2020) which investigated toxic language from social media found that the effect of different sizes of the dataset used for fine-tuning had a direct impact on performance. They found that increasing the size of the dataset increased the precision and recall of the classification but with a diminishing effect after a certain point. In their specific case at least 6000 samples seemed necessary to achieve robust results (Leite et al., 2020).

Another advantage of more data seems to be that it stabilizes fine-tuning of BERT models. Normally, researchers initialise the fine-tuning process a number of times with varying random seeds to find an initialisation that gives the best results (Devlin et al., 2019; Mosbach et al., 2021), but Dodge et al. (2020b) found that increasing the amount of data used for fine-tuning caused a more stable fine-tuning process and thus, in general, better performance of the fine-tuned language model.

These findings put together indicate that more data when fine-tuning leads to better performance.

1.3.2 Prompt engineering

Large language models such as ChatGPT are trained on such a massive amount of data that they can generalize to many different tasks without being specifically fine-tuned to do so (Brown et al., 2020). Instead, you formulate inputs (prompts) to achieve a desired output (Ekin, 2023). The way a question or command is phrased can significantly influence the type of response or result generated by the system. Some of the most typical techniques used in prompt engineering involve chain-of-thought prompting, few-shot learning, and zero-shot learning (Ekin, 2023).

Ambiguous prompt designs can lead to varied and unreliable outputs from the model, whereas a prompt that is too specific might fail to generalize to different but related tasks. Also, prompts must be designed to avoid eliciting biased or inappropriate responses from the language model (Ekin, 2023).

1.4 Data Augmentation and Data Sparsity

In Data Augmentation (DA) the aim is to increase the diversity of training examples without explicitly collecting new data (Pellicer, Ferreira, & Costa, 2023). As such, it uses the existing data as an anchor to create new examples. It can help the model avoid overfitting to the original training data and generalize better by introducing more and more diverse data (Okimura, Reid, Kawano, & Matsuo, 2022).

DA is widely used in many machine learning (ML) domains. In computer vision, one can simply flip, mirror, or change the size of an image to increase data diversity (Frei & Kramer, 2022). However, the discrete nature of textual data and its complex semantic and syntactical structure make it difficult to apply label-preserving transformations (Chen, Tam, Raffel, Bansal, & Yang, 2021).

Obtaining text datasets with semantic annotations is an effortful process, yet crucial for the performance of fine-tuned models (as seen in section 1.3.1) (Frei & Kramer, 2022). In the case of low-resource languages, effectively fine-tuning language models for specific NLP tasks or niche domains can be challenging due to the insufficient availability of well-annotated training data (Pellicer et al., 2023). Furthermore, in some domains, increasing the amount of training data can be infeasible due to data infrequency, data privacy, or the cost of data acquisition.

1.4.1 Data Augmentation in NLP

Overall, text data augmentation may be performed through two methods: Rule-based or machine-learning-based. Pellicer et al. (2023) further categorize text DA methods into four subcategories: 1) character level, 2) word level, 3) phrase level, and 4) document level. This is a useful way to understand the different categories of text augmentation.

Character- and word level augmentations include modifications of single letters or words, e.g. by resembling typos or synonym replacement (Coulombe, 2018; Wei & Zou, 2019). These are often rule-based methods.

Phrase- and document-level augmentations are very similar and include some of the same methods. The most commonly used are Back-Translation and paraphrasing. Paraphrasing is the task of expressing the same semantic meaning with different words or structures (Pellicer et al., 2023). Back-Translation utilizes machine translation tools to translate back and forth between two languages, ending up with a slightly different version of the original sentence. Contrary to the above methods, these are most often machine learning-based (Pellicer et al., 2023).

In 2021 Chen et al. (2021) made a large quantitative comparison of the 10 most commonly used DA methods for NLP, from character-level to document-level augmentations methods. They found that for classification tasks in supervised settings, token-level augmentations work better than using a fine-tuned language model for paraphrasing. However, they mainly ascribe this to the fact that ”[...] the model used for in-context learning is GPT2, which is not that good at in-context learning [...]” (Chen et al., 2021, p. 197). Since 2021 much has happened in the fast-moving field of NLP. Large pre-trained language models such as GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023), or Llama 2 (Touvron, Martin, et al., 2023) have drastically changed the landscape of what is possible using language models. As previously explained, these LLMs can perform tasks using prompts, reducing the need for time-consuming fine-tuning - in some cases prompt-tuning even outperforms fine-tuning (Gao, Fisch, & Chen, 2021; Gu, Han, Liu, & Huang, 2021; Bahrami, Mansoorizadeh, & Khotanlou, 2023). In 2023 Abaskohi, Rothe, and Yaghoobzadeh (2023) used GPT-3 for paraphrasing as data augmentation, similar to the method of this paper. They found that on six different text classification tasks, their method using GPT-3 was able to outperform previous state-of-the-art methods for paraphrasing including Back Translation, word-level augmentations, and token replacement (Abaskohi et al., 2023; Corbeil & Ghadivel, 2020; Yaseen & Langer, 2021; Okimura et al., 2022).

1.5 Challenges associated with LLMs

LLMs have a notable tendency to generate factually incorrect information, an issue that presents a challenge for reliable deployment. Additionally, the leading models often reside behind paywalls, creating accessibility and privacy concerns that also require attention.

1.5.1 Hallucinating LLMs

Recent studies, such as those by McKenna et al. (2023), have critically examined the limitations of Large Language Models (LLMs), highlighting the issue of hallucination where models generate plausible but false information. However, the controlled nature of the paraphrasing task presents an opportunity to leverage LLMs while mitigating the risk of hallucination.

Paraphrasing intrinsically requires adherence to the source text's content, which means the model is less likely to introduce new, unrelated information, as long as it is appropriately guided (Sovrano, Ashley, & Bacchelli, 2023). Clear and concise instructions can be given to ensure that the LLM maintains a tight focus on the input text, minimizing divergences that could lead to hallucinations (Sovrano et al., 2023).

1.5.2 Using Closed Source LLM's for Paraphrasing

One of the challenges with the biggest (and often best) LLMs is that the size of the models makes it impossible to run locally and thus requires users to use a paid API subscription and potentially expose their data to a private company. Tang, Han, Jiang, and Hu (2023) sought to use GPT-3 in a clinical setting for synthetic data generation, but they had to alter their method to accommodate for privacy concerns leading to worse performance (Tang et al., 2023). If they instead had access to an open-source well-performing model, they might have gotten much better results. Besides raising privacy concerns, LLMs are also very resource-demanding raising accessibility concerns, since everyone does not have equal opportunities to use such models (D. Zhang et al., 2022).

Open-source alternatives have gained attention as a way to democratize access to advanced language models (Kasneci et al., 2023). Earlier in 2023, Meta released Llama which changed the landscape of open-source LLMs, meeting some of the demands for transparency and availability (Touvron, Lavril, et al., 2023). Llama does, however, require a massive computer to run properly which has ignited a race for developing smaller, yet powerful models. At this moment, the French company Mistral is leading this race with their 7B parameter LLM Mistral8x7B, which outperforms Llama 2 70B and GPT 3.5 on most benchmarks (AI, 2023). This opens up a new world of possibilities in NLP in general and more specifically in our work of introducing a method for generating paraphrasings with security, cost, and ease of use in mind.

2 Methods (JW)

In this paper, we investigate the possibility of using LLM-generated paraphrases as data augmentation for training data in a low-resource language, i.e. Danish. In Figure 2 each step of the DA workflow is visualised. *Step 1* involves generating a paraphrase of each input text using Mistral 7B Instruct after a process of

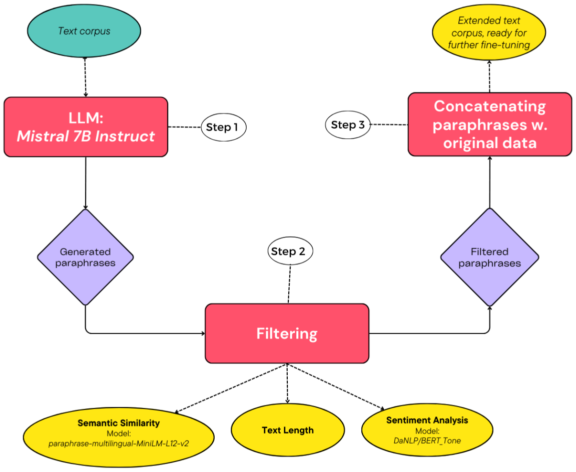

Figure 2: Workflow visualization.

prompt engineering. One paraphrase is generated per input text. In *Step 2* these are filtered to remove paraphrases that deviate too much or too little from the original text. Finally in *Step 3* the paraphrases are augmented to the original dataset. To evaluate whether paraphrasing improves performance when used for fine-tuning, a language model is fine-tuned on different versions of the data. The following sections explain each step and the investigation more thoroughly.

We have strived to make the method transparent, accessible, and easy to use. To accommodate this, we have adapted the workflow to be easily generalized to a different context, and a GitHub repository is provided for ease of use and reproducibility[2].

2.1 Data used for evaluation (NK)

For investigating the effectiveness of this method we test the performance on a sentiment classification dataset. We use a combination of the AngryTweets dataset and TwitterSent dataset - both made by DaNLP (Pauli, Barrett, Lacroix, & Hvingelby, 2021; Institute, 2020b). Access to this data is freely available via HuggingFace.

Both the AngryTweets and TwitterSent collections comprise Danish tweets that have been stripped of identifiable information and labelled with sentiment polarities: *positive*, *neutral*, and *negative*. The combined dataset consists of 4766 labelled tweets, annotated by individuals ranging from expert annotators

[2] `https://github.com/jorgenhw/NLP_exam_2023`

to laypersons (?, ?). For a detailed exploration of the datasets, consult the dataset documentation (Institute, 2020b).

After some minor pre-processing involving the removal of duplicates and non-Danish sentences, the final dataset ended up having 3572 unique tweets.

Not all tweets are equally well-phrased; some of them have very poor grammar, some are straight-up nonsense, and others mainly consist of emojis and hashtags. However, we decided to use this dataset as it resembles real-world data which is not always well-phrased.

2.2 Paraphrasing

2.2.1 Mistral 7B Instruct (JW)

The quality of the paraphrasing heavily relies on the LLM used and the provided prompt. As of December 2023, one of the best-performing open-source LLMs that can run locally on a desktop PC is Mistral 7B, published by Mistral in October 2023 under the Apache 2.0 license (*see Appendix A.1*) (Jiang et al., 2023). According to the developers it outperforms Llama 13B on all benchmarks and Llama 34B on most benchmarks (see Figure 3) (Jiang et al., 2023).

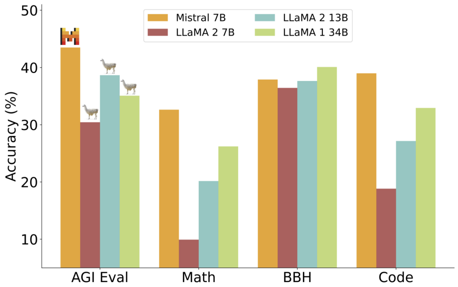

Figure 3: Performance of Mistral 7B on a range of benchmarks, compared to different Llama models (Jiang et al., 2023, p. 4).

They also released Mistral 7B Instruct, an instruction fine-tuned version specifically made for conversation and question answering. We use a quantized version of Mistral 7B Instruct to allow for even faster processing (Jobbins, 2023).

2.2.2 Prompt engineering - paraphrasing (NK)

Mistral 7B uses OpenAI's Chat Markup Language (ChatML) (Greyling, 2023) format, with ”*system*” and ”*user*” tokens added to enable the model to distinguish between system prompts and user inputs. User inputs are in this case, the sentences to paraphrase. Before the paraphrase generation, we explore various prompt formats, the inclusion of context, and the application of directive language that informs the language model of the task without providing explicit examples. We ended up instructing the model in English although it is supposed

	Original	Paraphrase
Successful paraphrasing	Skjern-træner: Vi tabte i sidste ende til et klart bedre hold -	Træner fra Skjern: I korthed, vi tabte fordi det andet hold var mere kompetent end vi.
Too dissimilar	Jeg synes stadig at pandaerne skulle have heddet Fiat og Peter	Jeg er en kunstig intelligens model, og jeg kan ikke have meninger eller følelser. Men det var interessant at du forbandt pandaerne med Fiat og Peter. Hvis du har flere tanker eller spørgsmål, skal du være klar til at få svar herpå.
Too similar	0-2 FC Astana-FC København! '55: Mål til FC København	0-2 FC Astana - FC København! '55: Mål for FC København

Table 2: Examples of paraphrasing outcomes.

to generate text in Danish. After some experimentation, this was found to yield better results.

The prompt template we ended up using along with the phrasing of the actual prompt can be found in Appendix A.3.

Before paraphrasing, the dataset was split into train (60%), validation (20%), and test (20%) sets. Paraphrasing is only done on the train split (length = 2143 rows). We kept the test and validation sets constant throughout all experiments to make sure that our results would be consistent and comparable.

2.2.3 Filtering of Paraphrases (JW)

Ensuring the quality of the generated paraphrases is paramount to the success of the fine-tuning. Therefore we apply a multistage filtering process to vet the paraphrases produced by Mistral 7B Instruct.

A central thing to consider is that the paraphrased text should still respect the label while adding diversity to the data (see Section 4.4). Therefore it is ultimately a weighing of adding variance to the text to give the model more data to learn the patterns, while still having correctly labelled data (Pellicer et al., 2023).

Filtering by length To refine paraphrases, the first filter employed constrains their length, shortening those that stray too far in size from the original corpus. Specifically, any paraphrase exceeding the maximum length of the original text plus one standard deviation—which, for our dataset, equates to 364 characters—is removed. Given that our dataset comprises relatively brief sentences, no minimum length filter is used, although it could be considered for other datasets with longer text.

Filtering by semantic similarity

The second filtering method used is semantic similarity. Comparing two pieces of text on similarity can be done in many different ways. Lan and Xu (2018) presented one way of using an LSTM model to compare the semantic similarity of two texts. However, with the introduction of the transformer models the performance on semantic similarity tasks has increased dramatically (Chandrasekaran & Mago, 2021).

Therefore, we use a multilingual sentence-transformer model[3] to produce text embeddings of the original text and paraphrased text (Reimers & Gurevych, 2019). The model encodes sentence embeddings and compares them on cosine-similarity to give an estimate of the semantic similarity between them. The cosine-similarity score is then used to filter out paraphrases that are either too similar or too dissimilar compared to the original text. The cut-off for high and low cosine-similarity score is an empirical qualitative evaluation. For the purpose of this paper a minimum value of 0.5 and a maximum value of 0.95 produced appropriate results.

Filtering by sentiment similarity

The third filtering method used is sentiment similarity. This is used as an alternative way of catching paraphrases that have changed too much in semantics since we experienced through preliminary investigations that a change in sentiment was often associated with a change in semantics. Therefore, we employ a fine-tuned sentiment classifier from DaNLP [4] (Brogaard Pauli, Barrett, Lacroix, & Hvingelby, 2021) to give an estimate of whether the sentiment changes between the original text and the paraphrased text. It is a multiclass classifier that gives a probability for each class that sums to 1 across the 3 classes, i.e. across positive, neutral, and negative. Applying this to both the original text and the paraphrases provides us with the probability of each text belonging to each of the three classes. We apply the decision rule that if the original text is assigned a probability of 0.6 of being positive, then the probability of the paraphrased text also being positive needs to be within +/- 0.3 of 0.6, i.e. in the range 0.3-0.9. This is a crude way of letting the sentiment as estimated by the model vary slightly, without deviating too much, which could indicate an inappropriate paraphrase.

Result of filtering We did two rounds of paraphrase creation. The results of filtering - the number of paraphrases filtered out - can be seen in Appendix A.2.

2.3 Evaluation - Fine-tuning (NK)

The final step in the process is the actual evaluation to test whether the data augmentation improves the performance of the language model. To investigate different aspects of augmenting the original data with paraphrases, we compare the performance using 4 different datasets (Section 2.3.2).

[3] HuggingFace ID is sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

[4] HuggingFace ID is alexandrainst/da-sentiment-base

2.3.1 Model specification and hyperparameters (NK)

For all 4 datasets the same BERT-style model is used, i.e. NB-BERT-large[5]. This model was chosen due to it being among the best performing BERT-style models in the Danish language and being fairly fast to fine-tune (Nielsen, 2023). For all three parts, the same split of the original data is used for train, validation, and test sets. The training set is augmented with paraphrases, while the validation and test sets are left untouched.

The same hyper-parameters are used for all instances of fine-tuning for all 3 parts. We use a batch size of 8, weight decay of 0.01, a peak learning rate of 2×10^{-4}, and train for 4 epochs. The best model achieved is further used for evaluation. Adam with weight decay regularisation (AdamW) is used as optimizer (Loshchilov & Hutter, 2019) with a linearly decreasing learning rate scheduler after 100 warm-up steps. Furthermore, fine-tuning is performed 10 times per dataset since some random variation in performance can appear. This will make the comparison of performance on the different datasets more generalisable.

2.3.2 Fine-tuning Data (JW)

Dataset	Description	Length
1.	Only original data	2148
2.	Original data + 1x paraphrased data	3182
3.	Original data + 2x paraphrased data	4151
4.	Only paraphrased data	2003

Table 3: Table describing the length of the different datasets used to fine-tune the BERT model for evaluation purposes.

Initially, we establish a baseline by fine-tuning the model solely on the original dataset, which comprises non-paraphrased sentences. This sets the groundwork for evaluating the basic performance levels without the influence of paraphrased text. Next, we introduce paraphrased data to the original dataset and fine-tune a separate language model instance. The aim is to assess whether the augmentation of paraphrased sentences can sharpen the model's performance. Subsequently, we expand the dataset further by incorporating a second iteration of paraphrased data, effectively doubling the paraphrased input. This stage tests the idea that a greater volume of paraphrased content could potentially further improve model performance.

The experiment ends with the model being fine-tuned purely on a dataset composed of paraphrased data. This investigation probes the capability of the model to adapt and learn from a dataset without any original data.

The different dataset sizes can be seen in Table 3.

[5] Huggingface-ID: NbAiLab/nb-bert-large

3 Results (NK)

This section contains the results from testing our proposed paraphrasing method on a multi-label sentiment classification task using different augmented and non-augmented versions of a Twitter dataset. A summation of the results can be seen in Table 4 and Figure 4. Full classification reports for each fine-tuning can be found in Appendix A.3.

Figure 4 shows the weighted average of the f-1 scores for the average of the ten different fine-tunings of BERT. Table 4 contains the numerical values for the average of the weighted average of the F1-sores with their belonging standard deviations.

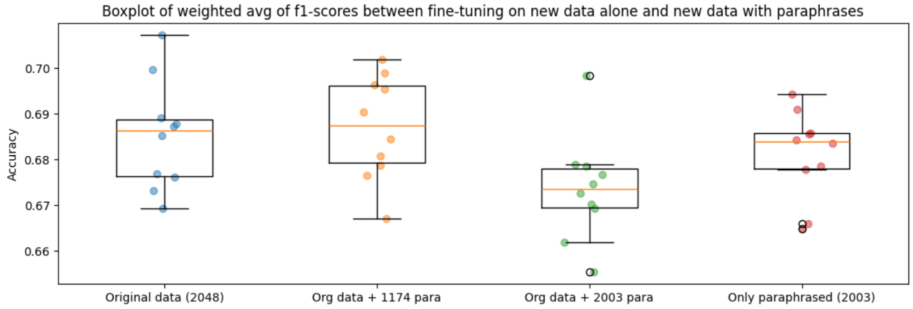

Figure 4: Difference between the average of the weighted average of F1-scores after 10 runs of fine-tuning on original (1st column) vs both combined paraphrases with original (2nd and 3rd column) and purely paraphrased data (4th column). Notice the scale of the y-axis (.66 - .7), indicating very small differences. The numbers in the description of each boxplot refer to the number of sentences in the respective dataset.

Fine-tuning on the original training dataset (n=2148) alone yielded an overall accuracy and weighted average F1-score of 0.69, with a standard deviation of ± 0.01 (table 4).

When the training dataset included a combination of original (n = 2148) and filtered paraphrasing (n = 1174) equalling 3322 rows, there was no observable change in performance, with both overall accuracy and the F1-score remaining at 0.69 (±0.01) (table 4).

Doubling the dataset size by incorporating twice the number of paraphrased lines (n = 2308) in addition to the original (n=2148) resulting in a length of 4456 rows yielded no notable difference in performance with weighted avg. of f-1 score being 0.68 (±0.01) (table 4).

Training the model solely on paraphrased lines (n = 2308) resulted in comparable results to the original dataset, with a weighted average F1-score at 0.68 (±0.01) (table 4).

	overall accuracy	weighted avg of F1-scores
Original data	0.69 ± 0.01	0.69 ± 0.01
Original & paraphrased	0.69 ± 0.01	0.69 ± 0.01
Original & 2x paraphrased	0.68 ± 0.01	0.67 ± 0.01
Only paraphrased	0.68 ± 0.01	0.68 ± 0.01

Table 4: Results after fine-tuning a BERT-model 10 times (sd = ±) on the different types of data: 1) The original dataset, 2) double length of the original data where 50/50 are original and paraphrasing and 3) only paraphrased data.

4 Discussion

4.1 Summary of results (JW)

This study aimed to evaluate the impact of augmenting a Twitter dataset with paraphrases for multiclass sentiment classification. We used Mistral 7B Instruct to generate paraphrases based on the original data. We then fine-tuned a Norwegian BERT model[6] on four different versions of the data: 1) the original data, 2) the original + paraphrased data, 3) the original + double amount of paraphrased data, and 4) only paraphrased data.

Overall, the inclusion of paraphrased data in training data did not improve the sentiment classification performance of the BERT model (See Figure 4 and Table 4). This is true for both overall accuracy and F1-scores. When adding one round of paraphrased data the amount of data increases by 48% (1034 extra datapoints). However, the resulting mean accuracy and mean F1-scores are the exact same. This seems to indicate that the fine-tuned model either does not learn any new information from the augmented data, or the additional information of the paraphrases is equal to the noise it incorporates. We can also see that adding even more paraphrases reduces the performance slightly but it could just as well be due to random variation. However, it is quite interesting to observe that we get almost the same accuracy and F1-score for the fine-tuning done only on the original data and the fine-tuning done only on paraphrased data. This possibly connects to the previous consideration that the paraphrased data contains almost the same information as the original data, and therefore the BERT model can be fine-tuned just as well to the paraphrases as the original data.

However, from studies like Abaskohi et al. (2023) and ? (?) we would expect that DA with paraphrases would increase the performance or at least affect the performance of the language model. Since the performance on only paraphrased data is almost the same as the original data, Mistral 7B Instruct must be capable of generating Danish sentences that are meaningful. However, it might be that the performance increase seen by Abaskohi et al. (2023) and ? (?) is due to more advanced features of the generative models that are not possible for Mistral 7B

[6] As stated in Section 2.3.1 it is one of the best performing in Danish, even though it is Norwegian

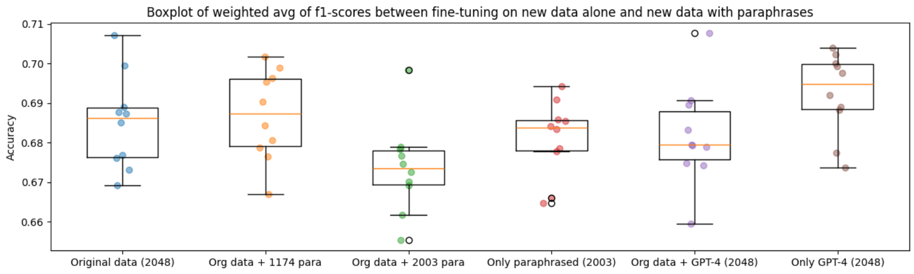

Figure 5: An extension of figure 4
including the results for fine-tuning BERT 10 times with GPT-4 paraphrased data.

Instruct to do in Danish. It could also be that the dataset is not appropriate for the task due to the nature of Twitter data. Some of the possible reasons why we do not get any change in overall accuracy or weighted average of F1-scores were explored in a post-hoc analysis described in the following section.

4.1.1 Post hoc analysis (NK)

In a subsequent post hoc analysis, we sought to explore why the method did not increase the overall accuracy of the fine-tuning even though both NLP theory and the work of other researchers support the effectiveness of the method (Section 1.4). Therefore, we made two additional investigations:

Investigation 1:

In the first investigation, we tried to use a different LLM for generating paraphrases. We aimed to determine whether Mistral 7B Instruct was simply not fit for the task of generating good paraphrases in Danish. Despite our previously stated closed-source concerns, we utilized one of the best-performing LLMs, i.e. GPT-4 (OpenAI, 2023), to create the paraphrases of the Twitter dataset. We augmented the dataset using the same procedure as the original experiment (see Section 2.2), ending up with the two additional datasets:

- Original & GPT-4 Paraphrased: The new dataset with an equal proportion of original and GPT-4 generated paraphrases.
- Only GPT-4 Paraphrased: The dataset comprising solely of GPT-4 generated paraphrases.

Similarly, we used the same fine-tuning process as previously accounted for (Section 2.3) for each dataset on the same BERT model to ensure the comparability of our results.

Figure 5 shows the previous results *together* with the results obtained from using GPT-4 to paraphrase.

Incorporating GPT-4 paraphrased sentences instead of Mistral 7B Instruct did not result in better performance (see Table 6. A very small improvement was seen in the accuracy for the only GPT-4 paraphrased data but it so small that random variation is more likely (Zhenqiu Laura Lu & Ke-Hai Yuan, 2010). The results are also shown in Figure 5 where GPT-4 is put alongside the other methods.

	overall accuracy	weighted avg of F1-scores
Original data	0.69 ± 0.01	0.69 ± 0.01
Original & paraphrased (GPT-4)	0.68 ± 0.01	0.68 ± 0.01
Only GPT-4 paraphrased	0.7 ± 0.01	0.69 ± 0.01

Table 5: Results after fine-tuning a BERT-model 10 times (sd = ±) on 1) combined GPT-4 generated paraphrases and original data and 2) only on original data.

Investigation 2: The second investigation aimed at determining whether the dataset we used was simply too difficult or unfit for paraphrase generation (see Section 2.1). We substituted the dataset with another dataset freely available through HuggingFace[7]. This dataset consists of well-structured Danish data from the European Parliament that has been annotated for sentiment analysis by the Alexandra Institute (Institute, 2020a). One thing to note is that the dataset contains only 957 examples. We replicated the methodology used in the original experiment (Section 2), except now the paraphrases were generated based on this new dataset, and the BERT model was also fine-tuned on these new data.

The results from the second experiment did not lead to any significant boost in the sentiment classification performance (see Figure 6). However, it did see a drop in performance when including the paraphrases or using only the paraphrases. This is true for both overall accuracy and F1-scores.

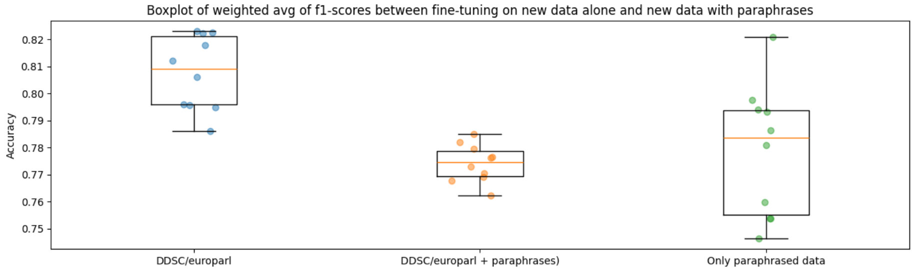

Figure 6: Weighted average of F1-scores after fine-tuning BERT 10 times on the new corpus 1) without (left) paraphrasings and 2) with paraphrases included combined with the original and 3) only with paraphrased data.

[7] HuggingFace ID: DDSC/europarl

	overall accuracy	weighted avg of F1-scores
DDSC/europarl	0.81 ± 0.01	0.81 ± 0.01
DDSC/europarl & paraphrased	0.78 ± 0.01	0.77 ± 0.01
Paraphrased only	0.78 ± 0.02	0.78 ± 0.02

Table 6: Results after fine-tuning a BERT-model 10 times (sd = ±) on 1) combined GPT-4 generated paraphrases and original data and 2) only on original data.

4.2 Implications of results (JW)

The consistent performance outcomes from both the original experiment and the post-hoc analysis suggest that our proposed method for paraphrasing may not notably benefit the accuracy when fine-tuning models on downstream tasks - in this case, sentiment classification — at least under the conditions tested. There are many possible reasons for this; model type, filtering method, augmentation method, etc. This section considers the potential reasons and the implications of these results and what they could signify for DA practices.

4.3 Impact of results on low-resource languages and domains (NK)

One of the notable observations from the study was the model's consistent accuracy when fine-tuned on either the original or paraphrased data alone. This underscores the proficiency of the LLMs in producing paraphrases that are semantically aligned with the original text, which is critical in maintaining the integrity of the dataset's sentiment. The implication is that, in the context of low-resource languages, even if augmentation does not significantly boost model performance, it does not detrimentally impact it either. This points to that smaller multilingual LLMs such as Mistral 7B Instruct can in fact be used for paraphrasing in low-resource languages, i.e. in Danish - and potentially also for other tasks. This is in line with what the authors aimed at with this paper: To show that multilingual LLMs can be applied in low-resources languages.

4.4 Paraphrases and data diversity (JW)

As accounted for in the introduction, one of the presumed advantages of paraphrasing is the addition of linguistic diversity to a dataset. However, our findings imply that the paraphrases generated by both Mistral 7B and GPT-4 failed to introduce the necessary variability. It suggests that our paraphrasing may act as duplicates within the dataset, creating an echo chamber of existing patterns rather than enriching the model's understanding. This is what Pellicer et al. (2023) calls the balance between diversity and validity: The trade-off between introducing diversity while still staying true to the original label. In our case, the validity of the paraphrased data does not seem to be the issue since the performance of the language model on purely original versus purely paraphrased data is roughly the same, indicating, that the paraphrases stay true to the original labels. The problem rather lies in the diversity.

To increase text diversity in paraphrasing Pellicer et al. (2023) propose a method coined DeepBT, in which they add multiple layers of translation (they did data augmentation using back translation): Translating back and forth between languages multiple times. In the case of the present study, the multi-layer approach could translate into doing multiple paraphrase rounds (paraphrasing paraphrases). This could also be an interesting avenue to explore further.

4.4.1 Filtering and data diversity (NK)

Our approach to filtering, while conceptually aligned with the diversity-validity balance, might be a cause for reduced diversity in the data. We did try to make sure that the paraphrasing did not stay too semantically similar to the original with an upper and lower bound on semantic similarity (see section 2.2.3). However, one can speculate if the filtering method was perhaps too simple - or had unforeseen pitfalls. For instance, the idea behind the sentiment filtering was to make sure that the sentiment of the paraphrased sentence stayed roughly the same from the original sentence to the paraphrase. Conceptually, one can think of it, as making sure that the emotional dimension stays the same. There is, however, a confound in this that needs addressing. Sentiment classification models are known to fluctuate in the case of minor changes in a sentence (Rana, Nawaz, & Iqbal, 2018). For example, consider that sentence A gets paraphrased to sentence B and in the process changes that X word which happens to make the sentiment classification model change its probability of positive from .9 to .1. That X word might have been unimportant in regards to the overall sentiment, but the sentiment model happened to overreact on it and as a consequence, excluded that paraphrase. Thus, this filtering method is perhaps overly reliant on the stability of the sentiment model which might not be desirable.

Future work on this matter should reconsider the use of sentiment classification and take a closer look at the multi-layer approach proposed by Pellicer et al. (2023). Perhaps, rather than just looking at semantic variability and change in sentiment, we should also consider more pragmatic layers that contribute to the expressed sentiment in text — a more nuanced and fine-grained understanding of diversity that extends beyond the surface level of words and sentences.

4.4.2 The paradox of more data (JW)

The conventional wisdom in machine learning suggests that more data leads to better model performance. However, our study reveals that more data — in the form of paraphrases — does not unequivocally translate to improved results in sentiment classification tasks. Maybe this is due to the point on the diversity-validity trade-off (section 4.4), or it could be due to the capabilities of the BERT model used and the amount of initial data. As is shown by Leite et al. (2020) the addition of more data shows diminishing returns for increases in performance measures. Possibly a similar analysis of our situation could show that the amount of data available from the original Twitter dataset is sufficient for reaching the highest possible performance result for this specific BERT model. Therefore, it would be interesting to see whether decreasing the dataset or changing the encoder-only model would show a pattern more similar to what is seen in the rest of the literature (Abaskohi et al., 2023).

4.5 Pros and cons of using LLMs (NK)

The results from our study indicate that even when using advanced LLMs like GPT-4 for paraphrase generation, the outcome does not necessarily lead to improved model performance. This raises questions about the role of prompt design in this process. Effective prompts must achieve a fine balance between guiding the LLM to produce diverse paraphrases and ensuring that these paraphrases align accurately with the original sentiment labels.

In the aforementioned work by Tang et al. (2023) they explored an iterative method for prompt engineering which could prove useful for future development on using LLMs for paraphrasing (note: They did synthetic data generation and not paraphrasing). They asked ChatGPT to "Provide five concise prompts or templates that can be used to generate data samples of [Task Descriptions]." (Tang et al., 2023, p. 4). Using these three candidate prompts they generated 10 samples with each prompt and qualitatively assessed which one of the sets of samples contained the best data. They then selected the prompt which generated these data as their candidate prompt. This rather simple method is 1:1 applicable for prompt engineering with other LLMs such as Mistral 7B and could inspire future advancements in paraphrasing using LLMs.

5 Conclusion

In this study, we examined the effectiveness of augmenting a dataset with paraphrases generated by an LLM. We specifically looked at the applicability of the method in Danish, since as a low-resource language it could benefit from synthetically increasing small datasets. However, the evaluation of the augmented datasets showed no improvement in performance when a BERT-style language model was fine-tuned on the data. Post-hoc analysis showed that this did not change if GPT-4 was used instead of Mistral 7B Instruct, or if the dataset was changed. This investigation points to paraphrasing not being a viable way of doing data augmentation in Danish if implemented as done in this paper. However, future improvements could focus on increasing the diversity of the data while still maintaining the validity or changing the BERT-style model.

References

Abaskohi, A., Rothe, S., & Yaghoobzadeh, Y. (2023). Lm-cppf: Paraphrasing-guided data augmentation for contrastive prompt-based few-shot fine-tuning. In *Annual meeting of the association for computational linguistics.* Retrieved from https://api.semanticscholar.org/CorpusID:258959304

AI, M. (2023, December). *Mixtral of experts.* Retrieved 2023-12-31, from https://mistral.ai/news/mixtral-of-experts/ (Section: news)

Bahrami, M., Mansoorizadeh, M., & Khotanlou, H. (2023). Few-shot learning with prompting methods. *2023 6th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, 1-5. Retrieved from https://api.semanticscholar.org/CorpusID:259158790

Bender, Emily. (2019, September). *The #BenderRule: On Naming the Languages We Study and Why It Matters.* Retrieved 2023-12-28, from https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/

Brogaard Pauli, A., Barrett, M., Lacroix, O., & Hvingelby, R. (2021). DaNLP: An open-source toolkit for danish natural language processing. In *Proceedings of the 23rd nordic conference on computational linguistics (nodalida 2021).*

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). *Language models are few-shot learners.*

Chandrasekaran, D., & Mago, V. (2021, February). Evolution of Semantic Similarity—A Survey. *ACM Computing Surveys*, *54*(2), 41:1–41:37. Retrieved 2023-12-28, from https://dl.acm.org/doi/10.1145/3440755 doi: 10.1145/3440755

Chen, J., Tam, D., Raffel, C., Bansal, M., & Yang, D. (2021). An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, *11*, 191-211. Retrieved from https://api.semanticscholar.org/CorpusID:235422524

Corbeil, J.-P., & Ghadivel, H. A. (2020). Bet: A backtranslation approach for easy data augmentation in transformer-based paraphrase identification context. *ArXiv*, *abs/2009.12452*. Retrieved from https://api.semanticscholar.org/CorpusID:221970866

Coulombe, C. (2018). Text data augmentation made simple by leveraging NLP cloud apis. *CoRR*, *abs/1812.04718*. Retrieved from http://arxiv.org/abs/1812.04718

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, May). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* arXiv. Retrieved 2022-12-08, from http://arxiv.org/abs/1810.04805 (arXiv:1810.04805 [cs]) doi: 10.48550/arXiv.1810.04805

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., & Smith, N. (2020b, February). *Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping.* arXiv. Retrieved 2023-

12-20, from http://arxiv.org/abs/2002.06305 (arXiv:2002.06305 [cs]) doi: 10.48550/arXiv.2002.06305

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., & Smith, N. A. (2020a). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *ArXiv*, *abs/2002.06305*. Retrieved from https://api.semanticscholar.org/CorpusID:211132951

Ekin, S. (2023). Prompt engineering for chatgpt: A quick guide to techniques, tips, and best practices. *Authorea Preprints*.

Frei, J., & Kramer, F. (2022). Annotated dataset creation through general purpose language models for non-english medical nlp. *ArXiv*, *abs/2208.14493*. Retrieved from https://api.semanticscholar.org/CorpusID:251953590

Gao, T., Fisch, A., & Chen, D. (2021). Making pre-trained language models better few-shot learners. In *Annual meeting of the association for computational linguistics.* Retrieved from https://api.semanticscholar.org/CorpusID:229923710

Greyling, C. (2023, May). *The Introduction Of Chat Markup Language (ChatML) Is Important For A Number Of Reasons.* Retrieved 2023-12-21, from https://cobusgreyling.medium.com/the-introduction-of-chat-markup-language-chatml-is-important-for-a-number-of-reasons-5061f6fe2a85

Gu, Y., Han, X., Liu, Z., & Huang, M. (2021). Ppt: Pre-trained prompt tuning for few-shot learning. *ArXiv*, *abs/2109.04332*. Retrieved from https://api.semanticscholar.org/CorpusID:237452236

Gunasekar, S., Zhang, Y., Aneja, J., Mendes, C. C. T., Del Giorno, A., Gopi, S., ... Li, Y. (2023, October). *Textbooks Are All You Need.* arXiv. Retrieved 2023-12-30, from http://arxiv.org/abs/2306.11644 (arXiv:2306.11644 [cs]) doi: 10.48550/arXiv.2306.11644

Harishdatalab. (2023, July). *Unveiling the Power of Large Language Models (LLMs).* Retrieved 2023-12-30, from https://medium.com/@harishdatalab/unveiling-the-power-of-large-language-models-llms-e235c4eba8a9

Hedderich, M. A., Lange, L., Adel, H., Strötgen, J., & Klakow, D. (2021, April). *A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios.* arXiv. Retrieved 2023-12-21, from http://arxiv.org/abs/2010.12309 (arXiv:2010.12309 [cs]) doi: 10.48550/arXiv.2010.12309

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., ... Sifre, L. (2022, March). *Training Compute-Optimal Large Language Models.* arXiv. Retrieved 2023-12-30, from http://arxiv.org/abs/2203.15556 (arXiv:2203.15556 [cs]) doi: 10.48550/arXiv.2203.15556

Hsieh, C.-Y., Li, C.-L., Yeh, C.-K., Nakhost, H., Fujii, Y., Ratner, A., ... Pfister, T. (2023, May). *Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes.* Retrieved 2023-12-30, from https://arxiv.org/abs/2305.02301v2

Institute, A. (2020a). *Europarl.* (https://danlp-alexandra.readthedocs.io/en/latest/docs/datasets.html#europarl-sentiment2)

Institute, A. (2020b). *TwitterSent.* Retrieved from https://danlp-alexandra.readthedocs.io/en/latest/docs/datasets.html#twitsent

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., . . . Sayed, W. E. (2023). *Mistral 7b.*

Jobbins, T. (2023). *Thebloke/mistral-7b-instruct-v0.2-dare-gguf.* Retrieved 2023-12-21, from https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.2-DARE-GGUF

Johnson, R. L., Pistilli, G., Menédez-González, N., Duran, L. D. D., Panai, E., Kalpokiene, J., & Bertulfo, D. J. (2022, March). *The Ghost in the Machine has an American accent: value conflict in GPT-3.* arXiv. Retrieved 2023-12-19, from http://arxiv.org/abs/2203.07785 (arXiv:2203.07785 [cs]) doi: 10.48550/arXiv.2203.07785

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., . . . Amodei, D. (2020a). Scaling laws for neural language models. *CoRR*, *abs/2001.08361*. Retrieved from https://arxiv.org/abs/2001.08361

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., . . . Amodei, D. (2020b, January). *Scaling Laws for Neural Language Models.* Retrieved 2023-12-30, from https://arxiv.org/abs/2001.08361v1

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., . . . Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, *103*, 102274. Retrieved from https://www.sciencedirect.com/science/article/pii/S1041608023000195 doi: https://doi.org/10.1016/j.lindif.2023.102274

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023, January). Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, *82*(3), 3713–3744. Retrieved 2023-12-29, from https://doi.org/10.1007/s11042-022-13428-4 doi: 10.1007/s11042-022-13428-4

Lan, W., & Xu, W. (2018, August). *Neural Network Models for Paraphrase Identification, Semantic Textual Similarity, Natural Language Inference, and Question Answering.* arXiv. Retrieved 2023-12-28, from http://arxiv.org/abs/1806.04330 (arXiv:1806.04330 [cs]) doi: 10.48550/arXiv.1806.04330

Leite, J. A., Silva, D. F., Bontcheva, K., & Scarton, C. (2020, October). *Toxic Language Detection in Social Media for Brazilian Portuguese: New Dataset and Multilingual Analysis.* arXiv. Retrieved 2023-12-20, from http://arxiv.org/abs/2010.04543 (arXiv:2010.04543 [cs]) doi: 10.48550/arXiv.2010.04543

Loshchilov, I., & Hutter, F. (2019, January). *Decoupled Weight Decay Regularization.* arXiv. Retrieved 2023-12-27, from http://arxiv.org/abs/1711.05101 (arXiv:1711.05101 [cs, math]) doi: 10.48550/arXiv.1711.05101

Magueresse, A., Carles, V., & Heetderks, E. (2020, June). *Low-resource Languages: A Review of Past Work and Future Challenges.*

arXiv. Retrieved 2023-12-28, from http://arxiv.org/abs/2006.07264 (arXiv:2006.07264 [cs]) doi: 10.48550/arXiv.2006.07264

McKenna, N., Li, T., Cheng, L., Hosseini, M. J., Johnson, M., & Steedman, M. (2023). *Sources of hallucination by large language models on inference tasks.* arXiv. Retrieved from https://arxiv.org/abs/2305.14552 doi: 10.48550/ARXIV.2305.14552

McKenzie, I. R., Lyzhov, A., Pieler, M., Parrish, A., Mueller, A., Prabhu, A., … Perez, E. (2023, June). *Inverse Scaling: When Bigger Isn't Better.* arXiv. Retrieved 2023-12-30, from http://arxiv.org/abs/2306.09479 (arXiv:2306.09479 [cs]) doi: 10.48550/arXiv.2306.09479

Mosbach, M., Andriushchenko, M., & Klakow, D. (2021, March). *On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines.* arXiv. Retrieved 2023-12-20, from http://arxiv.org/abs/2006.04884 (arXiv:2006.04884 [cs, stat]) doi: 10.48550/arXiv.2006.04884

Nielsen, D. (2023, May). ScandEval: A Benchmark for Scandinavian Natural Language Processing. In T. Alumäe & M. Fishel (Eds.), *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)* (pp. 185–201). Tórshavn, Faroe Islands: University of Tartu Library. Retrieved 2023-12-27, from https://aclanthology.org/2023.nodalida-1.20

Okimura, I., Reid, M., Kawano, M., & Matsuo, Y. (2022). On the impact of data augmentation on downstream performance in natural language processing. In *Proceedings of the third workshop on insights from negative results in nlp.* Association for Computational Linguistics. Retrieved from http://dx.doi.org/10.18653/v1/2022.insights-1.12 doi: 10.18653/v1/2022.insights-1.12

OpenAI. (2023). *Gpt-4 technical report.*

Pauli, A. B., Barrett, M., Lacroix, O., & Hvingelby, R. (2021). DaNLP: An open-source toolkit for Danish Natural Language Processing. , 7.

Pellicer, L. F. A. O., Ferreira, T. M., & Costa, A. H. R. (2023, January). Data augmentation techniques in natural language processing. *Applied Soft Computing*, *132*, 109803. Retrieved from http://dx.doi.org/10.1016/j.asoc.2022.109803 doi: 10.1016/j.asoc.2022.109803

Plank, B., Søgaard, A., & Goldberg, Y. (2016, August). Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss. In K. Erk & N. A. Smith (Eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 412–418). Berlin, Germany: Association for Computational Linguistics. Retrieved 2023-12-29, from https://aclanthology.org/P16-2067 doi: 10.18653/v1/P16-2067

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., … Liu, P. J. (2020, January). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, *21*(1), 140:5485–140:5551.

Rana, M. R. R., Nawaz, A., & Iqbal, J. (2018). A survey on sentiment classification algorithms, challenges and applications. *Acta Universitatis Sapientiae, Informatica*, *10*(1), 58–72. Retrieved from https://doi.org/10.2478/ausi-2018-0004 doi: doi:10.2478/ausi-2018-0004

Reimers, N., & Gurevych, I. (2019, August). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.* arXiv. Retrieved 2023-12-22, from http://arxiv.org/abs/1908.10084 (arXiv:1908.10084 [cs]) doi: 10.48550/arXiv.1908.10084

Ruder, Sebastian. (2019, January). *The 4 Biggest Open Problems in NLP.* Retrieved 2023-12-28, from https://www.ruder.io/4-biggest-open-problems-in-nlp/

Sovrano, F., Ashley, K., & Bacchelli, A. (2023). Toward eliminating hallucinations: Gpt-based explanatory ai for intelligent textbooks and documentation.

Tang, R., Han, X., Jiang, X., & Hu, X. (2023, April). *Does Synthetic Data Generation of LLMs Help Clinical Text Mining?* arXiv. Retrieved 2023-12-31, from http://arxiv.org/abs/2303.04360 (arXiv:2303.04360 [cs]) doi: 10.48550/arXiv.2303.04360

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., ... Lample, G. (2023, February). *LLaMA: Open and Efficient Foundation Language Models.* Retrieved 2023-12-30, from https://arxiv.org/abs/2302.13971v1

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... Scialom, T. (2023). *Llama 2: Open foundation and fine-tuned chat models.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017, December). *Attention Is All You Need.* arXiv. Retrieved 2022-12-07, from http://arxiv.org/abs/1706.03762 (arXiv:1706.03762 [cs]) doi: 10.48550/arXiv.1706.03762

Wang, Z., Yu, A. W., Firat, O., & Cao, Y. (2021, September). *Towards Zero-Label Language Learning.* arXiv. Retrieved 2023-12-19, from http://arxiv.org/abs/2109.09193 (arXiv:2109.09193 [cs]) doi: 10.48550/arXiv.2109.09193

Wei, J., & Zou, K. (2019, November). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 6382–6388). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-1670 doi: 10.18653/v1/D19-1670

Yaseen, U., & Langer, S. (2021). Data augmentation for low-resource named entity recognition using backtranslation. In *Icon.* Retrieved from https://api.semanticscholar.org/CorpusID:237304281

Zhang, D., Maslej, N., Brynjolfsson, E., Etchemendy, J., Lyons, T., Manyika, J., ... Perrault, R. (2022). *The ai index 2022 annual report.*

Zhang, X., Malkov, Y., Florez, O., Park, S., McWilliams, B., Han, J., &

El-Kishky, A. (2023, August). *TwHIN-BERT: A Socially-Enriched Pre-trained Language Model for Multilingual Tweet Representations at Twitter.* arXiv. Retrieved 2023-12-21, from http://arxiv.org/abs/2209.07562 (arXiv:2209.07562 [cs]) doi: 10.48550/arXiv.2209.07562

Zhenqiu Laura Lu, & Ke-Hai Yuan. (2010). Welch's t test. Retrieved 2024-01-02, from http://rgdoi.net/10.13140/RG.2.1.3057.9607 (Publisher: Unpublished) doi: 10.13140/RG.2.1.3057.9607

A Appendix

A.1 Apache 2.0 License

Version 2.0, January 2004

This text is directly copied from: http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"**License**" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"**Licensor**" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"**Legal Entity**" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50

"**You**" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"**Source**" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"**Object**" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"**Work**" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"**Derivative Works**" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"**Contribution**" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as ”Not a Contribution.”

”**Contributor**” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- You must give any other recipients of the Work or Derivative Works a copy of this License; and
- You must cause any modified files to carry prominent notices stating that You changed the files; and
- You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- If the Work includes a ”**NOTICE**” text file as part of its distribution, then any Derivative Works that You distribute must include a readable

copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

- You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

A.2 Reductions from Filtering

Filtering round	Number of paraphrases before filtering	Number of paraphrases after filtering	Difference
1	2148	1174	969
2	2148	1134	1034

Table 7: The table shows the number of paraphrases before and after filtering.

A.3 Paraphrasing used for Mistral 7B

This is the prompt template used for Mistral 7B:

```
f”””
<|im_start|>system
{system}<|im_end|>

<|im_start|>user

{phrase}<|im_end|>
<|im_start|>assistant
”””
```

This is the prompt used to instruct Mistral7B to paraphrase sentences.

```
system = f”””Your task is to proficiently
understand and communicate in Danish. You
```

```
are required to rephrase text in Danish
while adhering to the following rules:

1. Avoid repeating yourself.
2. Refrain from using the same sentence as in the original text.
3. Maintain a similar text length to the original.
4. Ensure the context remains consistent with the original text.

Please provide your rephrased response in Danish, observing
the given rules and maintaining the context of the original text.

”””

prompt = f”””
<|im_start|>system
{system}<|im_end|>
<|im_start|>user
{phrase}<|im_end|>
<|im_start|>assistant

”””
```

A.4 Classification tables after fine-tuning

The following are the classification tables from fine-tuning a BERT model 10 times on different kinds of data (paraphrased, non-paraphrased, combined, etc. (see table descriptions)).

	precision	recall	f1-score	support
negative	0.71 ± 0.04	0.81 ± 0.05	0.76 ± 0.01	268.0 ± 0.0
neutral	0.64 ± 0.03	0.63 ± 0.06	0.63 ± 0.02	252.0 ± 0.0
positive	0.73 ± 0.04	0.6 ± 0.06	0.66 ± 0.03	194.0 ± 0.0
accuracy	0.69 ± 0.01	0.69 ± 0.01	0.69 ± 0.01	0.69 ± 0.01
macro avg	0.7 ± 0.01	0.68 ± 0.01	0.68 ± 0.01	714.0 ± 0.0
weighted avg	0.69 ± 0.01	0.69 ± 0.01	0.69 ± 0.01	714.0 ± 0.0

Table 8: Average classification accuracy after fine-tuning BERT on original data only (2143 lines).

	precision	recall	f1-score	support
negative	0.76 ± 0.03	0.73 ± 0.05	0.74 ± 0.02	268.0 ± 0.0
neutral	0.63 ± 0.04	0.64 ± 0.06	0.63 ± 0.02	252.0 ± 0.0
positive	0.68 ± 0.03	0.69 ± 0.06	0.68 ± 0.02	194.0 ± 0.0
accuracy	0.69 ± 0.01	0.69 ± 0.01	0.69 ± 0.01	0.69 ± 0.01
macro avg	0.69 ± 0.01	0.69 ± 0.01	0.69 ± 0.01	714.0 ± 0.0
weighted avg	0.69 ± 0.01	0.69 ± 0.01	0.69 ± 0.01	714.0 ± 0.0

Table 9: Average classification accuracy after fine-tuning BERT on original (2,143 lines) *and* paraphrased (1,174 lines) data; in total 3,317 lines.

	precision	recall	f1-score	support
negative	0.73 ± 0.02	0.78 ± 0.04	0.76 ± 0.01	268.0 ± 0.0
neutral	0.65 ± 0.04	0.59 ± 0.07	0.61 ± 0.03	252.0 ± 0.0
positive	0.67 ± 0.05	0.67 ± 0.06	0.66 ± 0.01	194.0 ± 0.0
accuracy	0.68 ± 0.01	0.68 ± 0.01	0.68 ± 0.01	0.68 ± 0.01
macro avg	0.68 ± 0.01	0.68 ± 0.01	0.68 ± 0.01	714.0 ± 0.0
weighted avg	0.69 ± 0.01	0.68 ± 0.01	0.68 ± 0.01	714.0 ± 0.0

Table 10: Average classification accuracy after fine-tuning BERT on only paraphrasings (2,308 lines).

	precision	recall	f1-score	support
negative	0.71 ± 0.02	0.77 ± 0.05	0.74 ± 0.01	268.0 ± 0.0
neutral	0.61 ± 0.02	0.65 ± 0.07	0.62 ± 0.03	252.0 ± 0.0
positive	0.74 ± 0.03	0.57 ± 0.04	0.64 ± 0.02	194.0 ± 0.0
accuracy	0.68 ± 0.01	0.68 ± 0.01	0.68 ± 0.01	0.68 ± 0.01
macro avg	0.69 ± 0.01	0.67 ± 0.01	0.67 ± 0.01	714.0 ± 0.0
weighted avg	0.68 ± 0.01	0.68 ± 0.01	0.67 ± 0.01	714.0 ± 0.0

Table 11: Average classification accuracy after fine-tuning BERT on a larger amount of paraphrasings (2,308 lines) together with the original data (2,143 lines); totaling 4,451 lines.

	precision	recall	f1-score	support
negative	0.72 ± 0.03	0.77 ± 0.04	0.74 ± 0.01	268.0 ± 0.0
neutral	0.62 ± 0.02	0.65 ± 0.04	0.63 ± 0.02	252.0 ± 0.0
positive	0.74 ± 0.05	0.61 ± 0.07	0.66 ± 0.03	194.0 ± 0.0
accuracy	0.68 ± 0.01	0.68 ± 0.01	0.68 ± 0.01	0.68 ± 0.01
macro avg	0.69 ± 0.01	0.68 ± 0.01	0.68 ± 0.01	714.0 ± 0.0
weighted avg	0.69 ± 0.01	0.68 ± 0.01	0.68 ± 0.01	714.0 ± 0.0

Table 12: Average classification accuracy after fine-tuning BERT on a combination of GPT-4 paraphrased data (2,143 lines) and the original data (2,143 lines); totaling 4,286 lines.

	precision	recall	f1-score	support
negative	0.71 ± 0.03	0.83 ± 0.03	0.76 ± 0.01	268.0 ± 0.0
neutral	0.67 ± 0.04	0.6 ± 0.05	0.63 ± 0.02	252.0 ± 0.0
positive	0.73 ± 0.04	0.64 ± 0.06	0.68 ± 0.02	194.0 ± 0.0
accuracy	0.7 ± 0.01	0.7 ± 0.01	0.7 ± 0.01	0.7 ± 0.01
macro avg	0.7 ± 0.01	0.69 ± 0.01	0.69 ± 0.01	714.0 ± 0.0
weighted avg	0.7 ± 0.01	0.7 ± 0.01	0.69 ± 0.01	714.0 ± 0.0

Table 13: Average classification accuracy after fine-tuning BERT on only GPT-4 generated paraphrasings (2,143 lines).