On Evaluation Protocols for Data Augmentation in a Limited Data Scenario

Frédéric Piedboeuf
Université de Montréal, RALI
frederic.piedboeuf@umontreal.ca

Philippe Langlais
Université de Montréal, RALI

Abstract

Textual data augmentation (DA) is a prolific field of study where novel techniques to create artificial data are regularly proposed, and that has demonstrated great efficiency on small data settings, at least for text classification tasks. In this paper, we challenge those results, showing that classical data augmentation (which modify sentences) is simply a way of performing better fine-tuning, and that spending more time doing so before applying data augmentation negates its effect. This is a significant contribution as it answers several questions that were left open in recent years, namely : which DA technique performs best (all of them as long as they generate data close enough to the training set, as to not impair training) and why did DA show positive results (facilitates training of network). We further show that zero- and few-shot DA via LLMs such as ChatGPT or LLama2 can increase performances, confirming that this form of data augmentation is preferable to classical methods.

1 Introduction

Data augmentation (DA) consists in generating artificial data points with the hope of improving the training of a model. In this paper, we focus on interpretable textual DA (methods that generate new sentences) for text classification in a limited data scenario, a practical setting of interest (Chen et al., 2021). Research generally finds that DA provides a great increase on small data settings (Chen et al., 2021; Kumar et al., 2021), a small increase in classification with medium datasets (up to 1000 examples) (Karimi et al., 2021; Liesting et al., 2021), and almost no increase on large datasets (Kobayashi, 2018; Yang et al., 2020; Okimura et al., 2022).

In this paper, we show that existing experimental protocols in DA studies are misleading. In particular, inadequate fine-tuning of the baseline (model trained without DA) from previous studies lead to an overestimation of the impact of textual DA, and training the models for longer results in an absence of gains with classical DA methods. We also consider more realistic protocols for DA evaluation, where we either don't have access to clean validation data, or where we adjust the training/validation split to better reflect real-world conditions. Lastly, we compare newer DA methods using ChatGPT (Ouyang et al., 2022) and Llama2 (Touvron et al., 2023) and show that the only thing that can consistently increase classification performance is generating data akin to external data (with zero or few-shot data generation), rather than generating new data from the current training distribution (by paraphrasing or modifying training sentences), as classical DA approaches do.

Overall, this paper covers several important contributions. The first and most important of them lies in showing that classical data augmentation (DA methods from before the advent of very Large Language Models, or LLMs) does not work on textual classification, and we also answer why previous studies showed positive results. These positive results have been a big question in recent years, as nobody could explain with satisfaction why generated sentences helped transformers learn better, the best hypothesis being that it brought some kind of regularization to the network (Feng et al., 2021; Queiroz Abonizio and Barbon Junior, 2020).

Our second contribution is a questioning of the use of the validation data in data augmentation for small data settings. DA research generally assumes the availability of clean validation samples, an often unrealistic setting. With the splits used in those studies, one finds themself with a few tens of data points for training the model and a few hundreds for validation. We consider better splitting and more realistic uses of data, showing that it is often more advantageous to use all available data as training data and to fine-tune for a longer time.

The third contribution is showing that by using LLMs to generate novel sentences (and not

Proceedings of the 31st International Conference on Computational Linguistics, pages 3428–3443
January 19–24, 2025. ©2025 Association for Computational Linguistics

just paraphrasing existing ones), the performance increases, but that this is still not as efficient as manually collecting and annotating novel data. We furthermore analyze two strategies for generating novel data, namely zero-shot data generation (using the task description) and 3-shot data generation (giving three examples to the LLM), showing that in most cases the 3-shot strategy largely outperforms zero-shot generation.

The last contribution joins a small but growing literature comparing LLMs in showing that ChatGPT outperforms Llama2 (Liu et al., 2023; Plátek et al., 2023; Guo et al., 2023). To our knowledge, we are the first to study this in the context of DA.

The paper is organized as such. Section 2 goes over the literature on interpretable textual DA. Sections 3 and 4 explain respectively the datasets and the DA methods we compare. Section 5 describes our evaluation protocols, then Section 6 presents our results. We discuss our work in Section 7.

2 Related Work

We may categorize DA techniques in three broad families : word-level operations, paraphrasing, and generative DA. We note that this is not an absolute categorization, and each of these families regroup several types of techniques. We also refer to techniques that were developed before LLMs as *classical DA*. Methods of this family modify the starting sentences in some ways, bringing variations to the dataset.

The first family does so by affecting individual words. The most seminal technique of this category is EDA (Easy Data Augmentation), which considers four operations : word substitution, word deletion, word swapping, and insertion of related vocabulary (Wei and Zou, 2019; Liesting et al., 2021). For insertion and substitution, EDA uses WordNet (Miller, 1998) to find synonyms of words from the input sentence. Another simple word-level technique is AEDA (An easier data augmentation technique) (Karimi et al., 2021), where random punctuation is inserted in between words of the sentence.

Other ways of replacing words have also been considered, such as the use of a pre-trained neural network to predict masked words (Kobayashi, 2018), using pre-trained embeddings to find words close in the embedded space (Marivate and Sefara, 2020), or using BERT/BART conditioned on the class to predict masked words or spans (Wu et al., 2019; Kumar et al., 2021) (named CBERT and CBART in this paper, conditional- BERT/BART). While easy to implement, those techniques tend to bring little diversity.

The second family of techniques acts at the sentence level, by considering the whole sentence for creating paraphrases. The seminal technique representing this family is Back-Translation (BT), in which a sentence is translated to another language and then back into English (Hayashi et al., 2018; Yu et al., 2018; Corbeil and Ghadivel, 2020; Edunov et al., 2018). The use of generative models for paraphrasing has also been considered, for example by encoding and decoding a sentence through a VAE (Variational Auto-encoder) (Mesbah et al., 2019; Yerukola et al., 2021; Nishizaki, 2017). Some other strategies that have been considered are modifying syntactic trees (Coulombe, 2018), fine-tuning BART on an in-domain corpus of paraphrases (Okur et al., 2022), using an off-the-shelf T5 model for paraphrasing (T5-Tapaco) (Piedboeuf and Langlais, 2023), or an LLM. such as ChatGPT. for paraphrasing (Fang et al., 2023).

Finally, generative methods aim to generate novel sentences from the same distribution as the training data. With an ideal generative strategy, this is equivalent to collecting new data, without the annotation cost associated to the collection process. Early studies of these techniques used GPT-2 (Kumar et al., 2020; Liu et al., 2020; Yang et al., 2020; Bayer et al., 2023) or VAEs (Qiu et al., 2020; Piedboeuf and Langlais, 2022), but recently focus has shifted to the use of LLMs, first with GPT3 (Yoo et al., 2021; Sahu et al., 2022), and then with ChatGPT for zero or small data generation (Møller et al., 2023; Ubani et al., 2023; Shushkevich and Cardiff, 2023; Sharma and Feldman, 2023).

Some recent works pertinent to this paper are the research of Kim et al. (2022); Zhu et al. (2023). Kim et al. (2022) question the training-validation splitting methodology in semi-supervised learning, showing that it is more efficient to fine-tune on augmented samples (created with DA) and use the original training sentences as validation data, instead of using a classical training/validation split. In Zhu et al. (2023), the authors note that weakly supervised learning approaches are evaluated by assuming the availability of clean validation samples, which is not often the case when working with small data. They notably develop novel methods for testing training under more realistic small data learning settings. In Section 5, we take inspiration of those studies to define our new protocols.

3 Datasets

We use five popular datasets to test the various data augmentation strategies : SST-2, Irony, IronyB, TREC6, and SNIPS. SST-2 (Socher et al., 2013) is a dataset of movie review classification with the binary classes of positive or negative. Irony and IronyB (Van Hee et al., 2018) are the binary and multiclass version of an Irony detection dataset. In the binary task (Irony), one must detect if tweets are ironic or not, and in the multiclass version one must as well determine which type of irony the tweet represents, if the tweet is ironic (between polarity clash, situational irony, and other irony). TREC6 (Li and Roth, 2002) is a task of question classification, where questions must be separated into six classes (abbreviation, description, entities, human beings, locations, and numeric values). Finally, SNIPS (Coucke et al., 2018) is a dataset of intent classification, where short commands have to be classified into different intents such as PlayMusic or GetWeather. Some characteristics of the datasets are available in Table 1.

Name	SST2	Irony	IronyB	TREC6	SNIPS
|classes|	2	2	4	6	7
sent. len.	19.3	13.7	13.7	10.2	9.3
|train|	6920	2683	2681	5452	13084
|val|	872	460	460	500	700
|test|	1821	3834	3832	492	700

Table 1: Characteristics of the classification tasks tackled in this study. The length of the sentences is defined by the number of white-space separated tokens.

Due to lack of computing budget[1] we leave the investigations of other tasks such as those considered in Chen et al. (2021) to future work.

4 Data augmentation methods

While DA methods are continuously being proposed, objective evaluation is difficult due to the lack of extensive comparison studies. We point the reader to Chen et al. (2021) for a literature review on DA methods, and Ding et al. (2024) for one specific to generative approaches. Here, we compare LLM based methods to classical approaches. All strategies are illustrated in Figure 1. Code and all hyperparameters are available in the additional material.

4.1 Classical methods

Kumar et al. (2021); Piedboeuf and Langlais (2023) compared several "classical" DA methods, from which we select three families of strategies that have been shown to be efficient. Concretely, we test word-manipulation methods (EDA, AEDA), conditional contextual based methods (CBERT, CBART), and paraphrase based methods (T5, BT).

EDA (multiple operations on words) and AEDA (insertion of punctuation) can be implemented simply and are resource efficient. Following experiments and results from the literature, we affect 10% of the words of a sentence in EDA, and use the formula given in Karimi et al. (2021) to calculate the number of punctuation signs to insert for AEDA.[2]

CBERT and CBART are more involved methods that leverage the masked words prediction task to generate new words conditionally on the class. Concretely, we mask words, prepend the class to the sentence (with a separation token), and fine-tune the model to predict the masked words. Generation of sentences follow the same process. The difference between CBERT and CBART is that the latter can predict *spans* of words, allowing it more flexibility in the generated sentences.

BT and T5-Tapaco aim to produce paraphrases of the original sentences to bring diversity to the training set. In BT, paraphrases are generated by translating the sentence into a second language and then back into English, and we use WMT[3] with German as a pivot language, which has shown good performances in the past (Edunov et al., 2018). T5-Tapaco makes use of `T5-small-Tapaco`[4], which is a T5 model fine-tuned on the TaPaCo paraphrase corpus (Scherrer, 2020), allowing us to directly generate paraphrases of sentences.

4.2 Large Language Models

LLMs are ubiquitous in NLP, and have rapidly gained traction in data augmentation. Here, we

[1] As we carefully fine-tune every baseline and run every experiment 15 times, this demands a significant time investment.

[2] EDA works by randomly selecting one of four operations (insertion of related words, word swapping, word deletion, and word substitution) and applying it to a percentage of the words of the sentence. AEDA works by simply inserting random punctuations (among "?", ".", ";", ":", "!", and ",") into the sentence.

[3] https://huggingface.co/facebook/wmt19-De-en

[4] https://huggingface.co/hetpandya/t5-small-tapaco

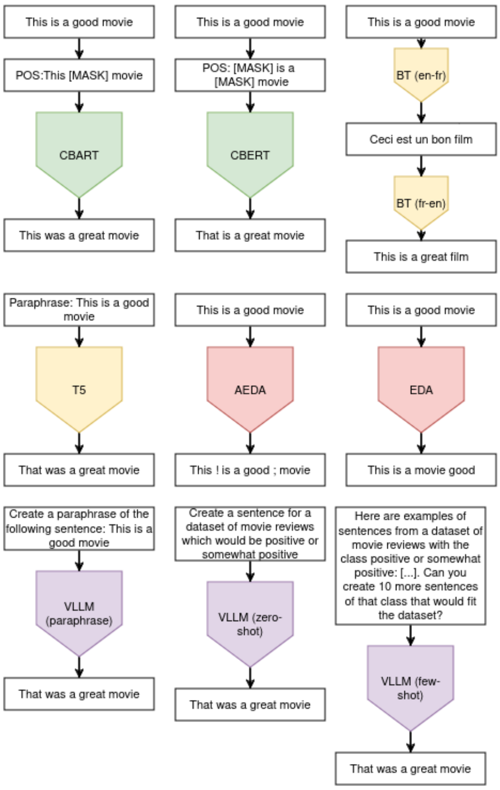

Figure 1: Strategies tested in this paper. Green algorithms are contextual unmasking methods, yellow are paraphrasing methods, red are word-manipulation methods, and purple are methods using LLMs.

test three standard strategies : paraphrasing sentences (P), zero-shot generation (ZS) by giving the LLMs a description of the task, and a 3-shot (3S) generation strategy where we give examples of the given class in addition to the description. Exact prompts used are provided in appendix B. We test two models, ChatGPT (GPT 3.5) and Llama2 (using `LLama-2-13B-Chat`) with the same prompts.

Note that we only use LLMs to generate new data with which we train a classifier, as with any other method we compare. Using LLMs directly to label examples may lead to better classification for some tasks, but we leave this for future investigations. In any case, training a classifier is a practical solution for many problems of interest, as well as a cheaper option to deploy.

4.3 Baselines

Finally, we implement three simple baselines. The first one consists in training the network without the use of DA (denoted "Baseline"). The second one is an idealized strategy (denoted "Perfect") where we fetch additional unused data from the training set to act as generated data. This gives an idea of the results obtainable should one collect and annotate data instead of using DA.

The last baseline is a strategy which we denote "Copy", and which artificially inflates the size of the training set by copying multiple times (according to a ratio parameter) the original data, without modification. If the only effect of DA is — as we suspect — to help fine-tune the network better, and the modifications brought by DA are not helpful for that, then the Copy strategy should be just as efficient as all other classical DA methods.

5 Experimental setups

In this section we describe the experimental setups for our two main experiments, that is to say the testing of DA methods with better fine-tuning, and the development of better experimental protocols for DA. Our focus in this paper is data augmentation for small data, for which we use starting size of 10 and 20, but we also report results for medium training sizes (500 and 1000) in Appendix C, a setting commonly explored in the literature. When subsampling to create the training set, we make sure to choose an equal number of data points for each class, by sampling more data if needed (e.g. the actual dataset size for TREC6 and starting size of 10 is 12; two sentences per class). There is no consensus regarding the ratio of generated-to-genuine examples, but we found in both the literature and experiments that smaller datasets benefited more from larger ratios, while for larger datasets, we did not observe improvements with ratios greater than one. Thus, we use a ratio of 10 for the small data settings and of one for the larger dataset sizes.[5] Finally, we use BERT as our classifier, fine-tuning it on our training set.

5.1 Better fine-tuning

To define our training protocol, we look at the literature to understand what is usually done. Piedboeuf and Langlais (2023) use variable patience[6] for fine-tuning, depending on the dataset and dataset size, between 5 and 20 epochs, and Kumar et al. (2021) use 100 epochs of warmup followed by 8 epochs of fine-tuning from which they select the

[5]Based on our main results, we can hypothesize that this is simply due to pretrained models needing more time to be fine-tuned correctly on smaller datasets than medium or large ones, and the larger ratio leads to a larger number of batches, meaning more training time.

[6]The number of epochs during which the validation performance doesn't increase before we stop the training process, and we then select the best model from previous epochs.

best model. Wei and Zou (2019) use early stopping with patience of 3 epochs on CNN and RNNs.

In contrast, we use a patience parameter of 50, using the validation set to find when to stop, and we fine-tune the learning rate and use label smoothing (Szegedy et al., 2015) (a mean of introducing noise to the labels for regularization) before applying data augmentation, using grid search.[7] We report accuracy for binary tasks and the macro-F1 for multiclass tasks.

5.2 More realistic uses of data

In academic papers, DA is often evaluated with the assumption that we have access to clean validation data, often in larger quantity than what is available for the training set. This validation data is used to fine-tune both the DA algorithm and the classifier, resulting in unrealistic settings for practitioners, who may not have access to validation data or want to use their data in better ways.

We aim to test more realistic settings with the validation data, inspired by Kim et al. (2022); Zhu et al. (2023), who study the same problem but in different settings (semi-supervised learning and weakly supervised learning) and then discuss the significance of the results for textual DA. While Kim et al. (2022); Zhu et al. (2023) findings are relevant to ours, our interest in the results differ. In both papers, the authors attempt to find the best way to use the available data. In our case, we test several settings we think are more realistic, but our interest is to know if DA is useful when there is little or no validation data for fine-tuning the augmentation method and the classifier.

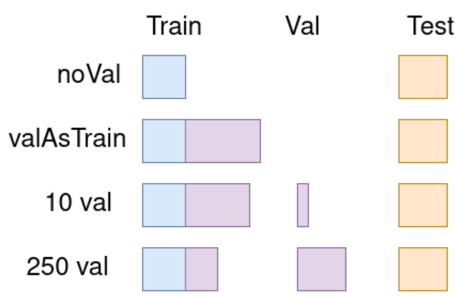

Figure 2: Graphical representation of the four settings we test for data augmentation on small data learning. Blue represent the original training set, purple the validation set, and yellow, the test set.

[7]While label smoothing was introduced in this paper as a mean of helping better learning of multiclass tasks, preliminary ablations experiments point that it raises the performance of both augmented and non-augmented dataset equally, and that the conclusions of this paper would be the same with only changing the number of epochs and hyper-parameter tuning.

We redefine the training/validation split, using different strategies illustrated in Figure 2. Firstly, we assume the validation data is not available (noVal), and we only have training data. In this setting, we train for a random number of epochs between 50 and 150, which is the range for which our models performed best in small data learning. This is inspired by the protocol of Zhu et al. (2023), where they randomly select a set of hyperparameters when no validation data is available. Secondly, we assume the presence of validation data but redefine the train/val. split, by either 1- using it all as training data and training for a random number of epochs between 4 and 8 (valAsTrain), or 2- keeping some of it as validation data (10, 250) to fine-tune both DA algorithms and the classifier (10 val/250 val).

6 Experiments

In what follows, we add the augmented material to the (small) training set and fine-tune a BERT-Base model[8] in a supervised way for each classification task, a solution which has been demonstrated efficient (Devlin et al., 2019).

6.1 On the need to better fine-tuning

We first show the inefficiency of the fine-tuning protocol used in past DA studies by reporting the results they obtain without DA and comparing it to our results (without DA as well). We attempt to replicate the starting sizes, changing only the fine-tuning protocol to see the difference a longer patience, label smoothing, and better fine-tuning does. Table 2 presents the results of our experiments compared to those from the literature. Since the use of label smoothing depends on the dataset, we integrate its use as a hyperparameter.

As we can see, carefully fine-tuning the classifier increases its performance significantly, gaining between 0 and 18.2 percentage pointsred. Those gains are actually larger than the ones reported by authors while deploying DA. This aligns well with our hypothesis that the gains observed in previous studies stem only from giving more time for the model to learn. Further, if the DA strategies involve simple transformations of the sentences, it is likely that BERT is not learning new information from these transformations, and that the same result will be achieved with the Copy strategy, which we analyze in the next section.

[8]We leave the study of other classifiers for future work.

	Reported	Ours	Diff
Kumar+Ubani SST-2	52.9	60.6	7.7
Kumar+Ubani SNIPS	48.6	66.8	18.2
Kumar+Ubani TREC6	79.4	91.6	12.2
Piedboeuf SST-2	87.7	87.7	0
Piedboeuf Irony	65.6	67.0	1.4
Piedboeuf IronyB	42.4	43.9	1.5
Piedboeuf TREC6	81.0	84.5	3.5

Table 2: Fine-tuning comparison from the literature and in this paper. All results are from training BERT without data augmentation, showing the difference made by using a longer patience as well as label smoothing.

6.2 On the inefficiency of classical DA

We now turn to DA algorithms, showing that better fine-tuning leads to essentially useless (classical) DA algorithms. Indeed we argue that alleged gains from DA that were shown in the literature can be explained by inadequate training of the (initial) classifier. Results are presented in Table 3.

For the small data setting, we notice that, on average, only two strategies really outperform the baseline : zero-shot and 3-shot generation (both using LLMs), 3-shot generation outperforming zero-shot, but not on all datasets. We explore this phenomenon in Section 6.4. The results for medium sizes (500 and 1000) are presented in Appendix C, and are concordant with more recent literature showing a lack of improvement on such a setting.

An important thing to note is that while some algorithms perform statistically better than the baseline at times, so does the "Copy" strategy. As the duplication of data should bring no new information, this suggests that the training protocol in previous studies was suboptimal and that the gains observed are not due to data augmentation.

One avenue that could be explored to bridge the gap would be to use sampling with replacement while fine-tuning BERT, which should be equivalent to duplication of data.[9]

Figure 3 provides the results of the statistical tests we conducted. To simplify reading, we group together all tests between the same algorithms, meaning all tests between algorithm A and algorithm B are compiled together. Each entry in the heatmap is therefore composed of 10 t-tests (5 datasets times 2 starting sizes), and we report the percentage of t-tests for which distributions were found to be statistically different, counting only the entries for which the row algorithm performs better than the column algorithm. This gives, for each row, the measure of how many times it beats the algorithm in the column by a statistically significant margin.[10]

[9]There would still be minor differences as sampling with replacement would not represent each sample equally, but this difference should reasonably not have a large impact.

	Baseline	Perfect	Copy	EDA	AEDA	BT	CBERT	CBART	T5	GPT3.5-P	GPT3.5-ZS	GPT3.5-3S	Llama2-P	Llama2-ZS	Llama2-3S
Baseline	0	20	10	10	10	30	10	10	10	20	20	40	10	30	30
Perfect	70	0	70	70	70	70	70	70	70	70	70	80	80	80	80
Copy	20	10	0	0	10	20	10	0	0	30	10	40	20	20	30
EDA	20	10	20	0	10	20	0	0	0	10	20	40	20	10	20
AEDA	10	10	0	0	0	20	0	0	0	10	0	30	20	0	0
BT	20	10	0	0	10	0	10	10	0	10	10	30	10	10	20
CBERT	20	10	10	0	10	10	0	10	0	10	0	20	10	0	20
CBART	10	10	0	0	0	10	0	0	0	10	0	30	10	10	10
T5	20	10	0	0	0	20	10	0	0	10	0	40	10	10	20
GPT3.5-P	30	10	10	20	10	40	20	20	10	0	10	30	10	20	20
GPT3.5-ZS	70	10	60	50	50	50	60	50	50	60	0	40	60	50	30
GPT3.5-3S	60	10	60	60	60	60	60	60	60	60	40	0	60	60	50
Llama2-P	20	10	10	0	10	20	20	10	0	0	0	30	0	0	10
Llama2-ZS	30	10	20	20	20	30	20	20	20	30	0	20	40	0	30
Llama2-3S	40	0	40	40	40	40	40	40	40	40	20	20	40	40	0

Figure 3: Percent of times the row algorithm performs statistically better than the column algorithm, with a p-value threshold of 0.05 and using a two-tails paired t-test, and across the two small data settings (10/20).

From the statistical tests, we note that only three methods outperform the others by a significant and consistent margin : the perfect strategy, in which we simulate collecting and annotating external data, and the zero/k-shot generation with LLMs. All others perform equivalently and do not beat the baseline most of the time, or at least not by a significant margin. This suggests that the slight random variations we observe, which could be taken as a sign that some algorithms perform better than others, are due to randomness of the network and not a superior performance.

6.3 On more realistic uses of data

As mentioned, the second point of interest is to evaluate DA in a more realistic setting with regard to the use of validation data. We select the following algorithms : Copy, Perfect, EDA, AEDA, BT, CBERT, BART, T5, GPT3.5-ZS, and GPT3.5-3S. We use this limited selection for efficiency reasons, but also because our goal here is to establish whether DA helps at all, rather than knowing which DA technique works best. Finally, to keep in line with other DA studies, we use a ratio of 10 when the final training size is small (noVal) and of 1 for the other settings. Results are presented in table 4,

[10]We provide the code along with the full saved results for all experiments, as well as the code necessary to run all statistical tests. See Figure 4 in Appendix for medium data settings (500/1000).

	SST2	Irony	IronyB	Trec6	SNIPS	Average
Baseline	62.4/67.5	54.1/61.9	25.8/29.8	34.4/43.3	78.5/80.4	51.1/56.6
Perfect	82.3/85.8	62.4/58.7	19.3/33.7	72.5/80.6	94.0/94.9	66.1/70.7
Copy	61.0/68.1	54.4/57.1	**25.5/30.3**	36.4/49.3	77.7/83.4	51.0/57.6
EDA	63.4/69.0	54.9/**58.9**	23.8/30.1	35.8/48.0	76.6/83.2	50.9/57.8
AEDA	62.4/67.1	55.4/56.9	23.4/27.4	33.2/45.7	78.0/84.0	50.5/56.2
BT	60.2/66.8	55.1/58.0	24.7/27.9	29.1/51.5	75.0/82.6	48.8/57.4
CBERT	62.5/65.4	56.3/58.5	23.2/28.3	36.4/44.5	76.3/83.4	50.9/56.0
CBART	62.5/65.6	54.5/57.7	23.6/29.0	36.9/47.2	77.0/84.5	50.9/56.8
T5	64.1/67.4	56.3/58.3	24.1/28.1	34.3/48.5	76.7/84.1	51.1/57.2
GPT3.5-P	65.0/68.3	**56.6**/57.5	22.7/28.3	34.0/40.5	80.2/84.7	51.7/55.9
GPT3.5-ZS	82.0/**78.7**	56.4/56.1	23.5/24.6	40.0/48.1	84.8/88.7	57.3/59.2
GPT3.5-3S	**87.7**/76.1	49.2/53.6	20.9/24.6	**58.4/63.9**	**87.2/89.6**	**60.7/61.5**
Llama2-P	66.3/68.6	55.3/57.1	24.6/24.6	33.4/41.6	78.1/82.8	51.6/55.0
Llama2-ZS	74.8/74.0	54.3/57.2	22.2/24.1	37.3/47.3	78.1/84.1	53.4/57.3
Llama2-3S	63.3/64.4	54.1/54.8	21.0/27.2	48.6/55.8	83.3/88.2	54.0/58.1

Table 3: Average metric over 15 runs for the training set sizes of 10 (left) and 20 (right) with a ratio of 10. We report accuracy for binary tasks and macro-f1 for multiclass ones. STDs are between 0.6 and 3.0, depending on the dataset. Results for which the difference with the baseline was found to be statistically significant according to a t-test are underlined.

and several important observations can be made.

	noVal	valAsTrain	10 val	250 val
Baseline	47.9	75.8	64.6	64.7
Perfect	68.8	82.4	69.4	69.9
Copy	49.6	**78.6**	**66.3**	65.8
EDA	49.1	77.8	66.0	64.9
AEDA	49.4	77.6	66.0	**66.3**
BT	44.8	75.9	64.6	65.0
CBERT	49.7	76.8	65.3	63.9
BART	50.6	76.6	65.3	63.9
T5Par	49.4	77.2	65.3	64.6
GPT3.5-ZS	55.9	76.9	65.6	64.4
GPT3.5-3S	**57.2**	71.6	67.2	64.8

Table 4: Average metrics over the five datasets for different settings and DA strategies. The best result for each setting is in bold.

First, in almost all cases and contrary to the main results of this paper, data augmentation reveals itself to be useful. However, the fact that the "Copy" strategy often outperforms the other strategies reinforces our belief that the use of data augmentation is simply to facilitate fine-tuning the network.

Second, there is no difference in performance between having 10 and 250 data points as validation data, with the rest as training set. This could be related to diminished increase in model performance as dataset size augments, or point to a trade-off in validation/training size and that a validation set of 10 data points is too small to be of use.

Third, we confirm here that methods adding something akin to external data (GPT3.5-ZS and GPT3.5-3S) are the most useful techniques, but only on small training sizes. On larger ones, those methods do not bring any improvement whatsoever, most likely due to the general inefficiency of data augmentation in this setting.[11] We also note that while the performance of GPT3.5-ZS is higher than GPT3.5-3S for two settings, this is due exclusively to its poor performance on the Irony datasets, which we discuss in Section 6.4. For the other datasets, GPT3.5-3S outperforms GPT3.5-ZS by a large margin.

Finally, it seems that the best strategy overall when the total amount of available data is only a few hundred is to use all of them as training data, with the Copy strategy.[12] This outperforms by a

[11] This is consistent with the literature and our own experiments.

[12] Or, we can assume, a larger number of epoch.

large margin the strategies where we keep some data as validation data, most likely because these strategies overfit. As mentioned, however, there is a large gap between 10 and 250 data and it is possible that some amount of data in the middle would perform better. We leave this investigation for future work.

6.4 Analysis of DA with LLMs

Our main results show that in a small data setting, 3-shot generation was often more efficient than zero-shot generation, by as much as 18 percentage points in some cases. However, this does not hold for the two irony datasets, for which zero-shot outperforms 3-shot. In this section, we analyze the output of the algorithms, in the hope of understanding better why some prompt mechanisms work well on some datasets but not on others.

We first show examples of generated sentences in Table 5 and in Appendix A. For the sake of space, we only show generated sentences for the tasks of SST-2 and Irony, but we give all sentences generated by ChatGPT in the associated Github.

	Couldn't have asked for a better night.	
Non-Ironic	-P	Could not have hoped for a more perfect evening.
	-ZS	Wow, I absolutely love it when my phone battery dies in the middle of an important call. Just what I needed!
	-3S	Enjoying a relaxing day at the beach with clear skies and warm sunshine. #PerfectDay
	Work should be fun today :unamused_face:	
Ironic	-P	Today at work is expected to be unenjoyable :unamused_face:
	-ZS	Oh, how wonderful! Another day of unexpected surprises and setbacks. It's like the universe's way of keeping me on my toes!
	-3S	Declared I'd be more organized and lost my keys for the umpteenth time. I'm the queen of tidiness! :key::mag: #OrganizationQueen

Table 5: Generated sentences by the GPT3.5-based DA methods for the Irony dataset for one ironic and one non-ironic example.

.

By observing generated sentences, we see that ChatGPT has difficulty understanding the concept of Irony, as well as sticking to the more familiar language of Twitter. As such, adding more sentences doesn't help because the new sentences are too far from the training distribution to bring valuable information. Nevertheless, the 3-shot strategy does seem to bring the generated sentences closer to the training distribution, as we can at least observe that the new sentences contain hashtags and emojis. It is likely that by further fine-tuning the prompts, we would reduce the gap between the data distribution and the generated sentences.

7 Conclusion

In this paper, we test DA for sentence classification and show that in both medium and small data learning, performances of DA had been overestimated by the use of inadequate training protocols. Furthermore, while we have shown that classical DA methods are inefficient, we have also demonstrated that simulating external data collection with LLMs does improve performance, and future work should therefore focus on this rather than techniques modifying genuine sentences.

We want to emphasize that our results are only looking at *balanced classification tasks*, and furthermore only at short text classification using BERT as a classifier. While this seems a small field of study, it is one of the most popular in textual DA literature, making our findings significant.

Future work should also evaluate whether the protocols are adapted for fields close to interpretable DA. Notably non-interpretable DA, or DA for unbalanced data, might still be an efficient mean of increasing performances. Other textual tasks might also still benefit from DA, due to different goals. As an example, data augmentation for the task of keyphrase generation (Ray Chowdhury et al., 2022; Garg et al., 2023) aims to encourage the network to generate more keyphrases that are absent from the input. It is very plausible that data augmentation may be useful here.

Our findings are important as they answer many questions in textual DA which have been standing for years, namely which data augmentation algorithms is best (they all perform similarly except for data generation methods), why does it help (gives more time to the network to learn), and what makes a generated sentence informative (they don't bring contradictory information to the network). As noted, data generation remains one of the best way to perform data augmentation, if one can get the LLMs to generate data adequate for the training distribution.

8 Limitations

As already stated, this study has a specific, although important, scope. We didn't look at the impact of data augmentation on other languages than English, or other textual tasks, and while we performed fine-tuning to the best of our knowledge, it is possible that some combinations of hyperparameters that were not explored may change some of our conclusions. We also omitted an ablation study of the fine-tuning (patience, label smoothing, grid search) due to time limitation, but this would provide more insight into the mechanism of training BERT. Furthermore, our exploration of hyperparameters for ChatGPT (mainly the temperature) was limited, and we only used the Llama2-13b-chat model as a comparison, but a comparison of other sizes and also other LLMs should be included in future work, such as Llama3 (Meta, 2024), Gemini (Team et al., 2024), GPT-4 (OpenAI et al., 2024), and others. Finally, we only looked at BERT as a classifier, which is the most common one used in DA studies, but it would be pertinent to not only look at other pretrained transformers but also at the impact of DA on more classical models such as CNNs, RNNs, or even statistical methods (SVM, NB, etc).

On a broader scope, there is no guarantee our results hold for other textual tasks (such as question answering, explainability, or keyphrase generation), which have different structures. A final point is that we only compared ourselves to the studies of Kumar et al. (2021); Piedboeuf and Langlais (2023) to demonstrate the inefficiency of fine-tuning in past studies. While these are some of the most complete we found, it would be pertinent to repeat the experience with other papers.

References

Markus Bayer, Marc-André Kaufhold, Björn Buchhold, Marcel Keller, Jörg Dallmeyer, and Christian Reuter. 2023. Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. *International journal of machine learning and cybernetics*, 14(1):135–150. Publisher: Springer.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2021. An Empirical Survey of Data Augmentation for Limited Data Learning in NLP. *arXiv:2106.07499 [cs]*. ArXiv: 2106.07499.

Jean-Philippe Corbeil and Hadi Abdi Ghadivel. 2020. BET: A Backtranslation Approach for Easy Data Augmentation in Transformer-based Paraphrase Identification Context. ArXiv:2009.12452 [cs].

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces. ArXiv:1805.10190 [cs].

Claude Coulombe. 2018. Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs. ArXiv:1812.04718 [cs].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using LLMs: Data perspectives, learning paradigms and challenges. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1679–1705, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. ArXiv:1808.09381 [cs].

Yihao Fang, Xianzhi Li, Stephen W. Thomas, and Xiaodan Zhu. 2023. ChatGPT as Data Augmentation for Compositional Generalization: A Case Study in Open Intent Detection. ArXiv:2308.13517 [cs].

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.

Krishna Garg, Jishnu Ray Chowdhury, and Cornelia Caragea. 2023. Data Augmentation for Low-Resource Keyphrase Generation. ArXiv:2305.17968 [cs].

Yue Guo, Zian Xu, and Yi Yang. 2023. Is ChatGPT a Financial Expert? Evaluating Language Models on Financial Natural Language Processing. ArXiv:2310.12664 [cs].

T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, and K. Takeda. 2018. Back-Translation-Style Data Augmentation for end-to-end ASR. In

2018 IEEE Spoken Language Technology Workshop (SLT), pages 426–433.

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.

Hazel H. Kim, Daecheol Woo, Seong Joon Oh, Jeong-Won Cha, and Yo-Sub Han. 2022. ALP: Data Augmentation Using Lexicalized PCFGs for Few-Shot Text Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10894–10902. Number: 10.

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.

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.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2021. Data Augmentation using Pre-trained Transformer Models. ArXiv:2003.02245 [cs].

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics -*, volume 1, pages 1–7, Taipei, Taiwan. Association for Computational Linguistics.

Tomas Liesting, Flavius Frasincar, and Maria Mihaela Truşcă. 2021. Data augmentation in a hybrid approach for aspect-based sentiment analysis. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, pages 828–835, New York, NY, USA. Association for Computing Machinery.

Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 2020. Data Boost: Text Data Augmentation Through Reinforcement Learning Guided Conditional Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9031–9041. ArXiv:2012.02952 [cs].

Zhengliang Liu, Yiwei Li, Peng Shu, Aoxiao Zhong, Longtao Yang, Chao Ju, Zihao Wu, Chong Ma, Jie Luo, Cheng Chen, Sekeun Kim, Jiang Hu, Haixing Dai, Lin Zhao, Dajiang Zhu, Jun Liu, Wei Liu, Dinggang Shen, Tianming Liu, Quanzheng Li, and Xiang Li. 2023. Radiology-Llama2: Best-in-Class Large Language Model for Radiology. ArXiv:2309.06419 [cs].

Vukosi Marivate and Tshephisho Sefara. 2020. Improving Short Text Classification Through Global Augmentation Methods. In *Machine Learning and Knowledge Extraction*, Lecture Notes in Computer Science, pages 385–399, Cham. Springer International Publishing.

Sepideh Mesbah, Jie Yang, Robert-Jan Sips, Manuel Valle Torre, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. 2019. Training Data Augmentation for Detecting Adverse Drug Reactions in User-Generated Content. 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 2349–2359, Hong Kong, China. Association for Computational Linguistics.

Meta. 2024. Meta Llama 3.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Anders Giovanni Møller, Jacob Aarup Dalsgaard, Arianna Pera, and Luca Maria Aiello. 2023. Is a prompt and a few samples all you need? Using GPT-4 for data augmentation in low-resource classification tasks. ArXiv:2304.13861 [physics].

H. Nishizaki. 2017. Data augmentation and feature extraction using variational autoencoder for acoustic modeling. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1222–1227.

Itsuki Okimura, Machel Reid, Makoto Kawano, and Yutaka Matsuo. 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*, pages 88–93, Dublin, Ireland. Association for Computational Linguistics.

Eda Okur, Saurav Sahay, and Lama Nachman. 2022. Data Augmentation with Paraphrase Generation and Entity Extraction for Multimodal Dialogue System. ArXiv:2205.04006 [cs].

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 Technical Report. ArXiv:2303.08774 [cs].

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. ArXiv:2203.02155 [cs].

Frédéric Piedboeuf and Philippe Langlais. 2022. Effective Data Augmentation for Sentence Classification Using One VAE per Class. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3454–3464, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Frédéric Piedboeuf and Philippe Langlais. 2023. Is ChatGPT the ultimate Data Augmentation Algorithm. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Ondřej Plátek, Vojtěch Hudeček, Patricia Schmidtová, Mateusz Lango, and Ondřej Dušek. 2023. Three Ways of Using Large Language Models to Evaluate Chat. ArXiv:2308.06502 [cs].

Siyuan Qiu, Binxia Xu, Jie Zhang, Yafang Wang, Xiaoyu Shen, Gerard de Melo, Chong Long, and Xiaolong Li. 2020. EasyAug: An Automatic Textual Data Augmentation Platform for Classification Tasks. In *Companion Proceedings of the Web Conference 2020*, pages 249–252. Association for Computing Machinery, New York, NY, USA.

Hugo Queiroz Abonizio and Sylvio Barbon Junior. 2020. Pre-trained Data Augmentation for Text Classification. In *Intelligent Systems*, Lecture Notes in Computer Science, pages 551–565, Cham. Springer International Publishing.

Jishnu Ray Chowdhury, Seo Yeon Park, Tuhin Kundu, and Cornelia Caragea. 2022. KPDROP: Improving Absent Keyphrase Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4853–4870, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Gaurav Sahu, Pau Rodriguez, Issam H. Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data Augmentation for Intent Classification with Off-the-shelf Large Language Models. ArXiv:2204.01959 [cs].

Yves Scherrer. 2020. TaPaCo: A Corpus of Sentential Paraphrases for 73 Languages.

Ashwyn Sharma and David I Feldman. 2023. Team Cadence at MEDIQA-Sum 2023: Using ChatGPT as a Data Augmentation Tool for Classifying Clinical Dialogue. CLEF.

Elena Shushkevich and John Cardiff. 2023. Tudublin at CheckThat! 2023: Chatgpt for data augmentation. *Working Notes of CLEF*.

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.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the Inception Architecture for Computer Vision. ArXiv:1512.00567 [cs].

Gemini Team, Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry, Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, Luke Vilnis, Oscar Chang, Nobuyuki Morioka, George Tucker, Ce Zheng, Oliver Woodman, Nithya Attaluri, Tomas Kocisky, Evgenii Eltyshev, Xi Chen, Timothy Chung, Vittorio Selo, Siddhartha Brahma, Petko Georgiev, Ambrose Slone, Zhenkai Zhu, James Lottes, Siyuan Qiao, Ben Caine, Sebastian Riedel, Alex Tomala, Martin Chadwick, Juliette Love, Peter Choy, Sid Mittal, Neil Houlsby, Yunhao Tang, Matthew Lamm, Libin Bai, Qiao Zhang, Luheng He, Yong Cheng, Peter Humphreys, Yujia Li, Sergey Brin, Albin Cassirer, Yingjie Miao, Lukas Zilka, Taylor Tobin, Kelvin Xu, Lev Proleev, Daniel Sohn, Alberto Magni, Lisa Anne Hendricks, Isabel Gao, Santiago Ontanon, Oskar Bunyan, Nathan Byrd, Abhanshu Sharma, Biao Zhang, Mario Pinto, Rishika Sinha, Harsh Mehta, Dawei Jia, Sergi Caelles, Albert Webson, Alex Morris, Becca Roelofs, Yifan Ding, Robin Strudel, Xuehan Xiong, Marvin Ritter, Mostafa Dehghani, Rahma Chaabouni, Abhijit Karmarkar, Guangda Lai, Fabian Mentzer, Bibo Xu, YaGuang Li, Yujing Zhang, Tom Le Paine, Alex Goldin, Behnam Neyshabur, Kate Baumli, Anselm Levskaya, Michael Laskin, Wenhao Jia, Jack W. Rae, Kefan Xiao, Antoine He, Skye Giordano, Lakshman Yagati, Jean-Baptiste Lespiau, Paul Natsev, Sanjay Ganapathy, Fangyu Liu, Danilo Martins, Nanxin Chen, Yunhan Xu, Megan Barnes, Rhys May, Arpi Vezer, Junhyuk Oh, Ken Franko, Sophie Bridgers, Ruizhe Zhao, Boxi Wu, Basil Mustafa, Sean Sechrist, Emilio Parisotto, Thanumalayan Sankaranarayana Pillai, Chris Larkin, Chenjie Gu, Christina Sorokin, Maxim Krikun, Alexey Guseynov, Jessica Landon, Romina Datta, Alexander Pritzel, Phoebe Thacker, Fan Yang, Kevin Hui, Anja Hauth, Chih-Kuan Yeh, David Barker, Justin Mao-Jones, Sophia Austin, Hannah Sheahan, Parker Schuh, James Svensson, Rohan Jain, Vinay Ramasesh, Anton Briukhov, Da-Woon Chung, Tamara von Glehn, Christina Butterfield, Priya Jhakra, Matthew Wiethoff, Justin Frye, Jordan Grimstad, Beer Changpinyo, Charline Le Lan, Anna Bortsova, Yonghui Wu, Paul Voigtlaender, Tara Sainath, Shane Gu, Charlotte Smith, Will Hawkins, Kris Cao, James Besley, Srivatsan Srinivasan, Mark Omernick, Colin Gaffney, Gabriela Surita, Ryan Burnell, Bogdan Damoc, Junwhan Ahn, Andrew Brock, Mantas Pajarskas, Anastasia Petrushkina, Seb Noury, Lorenzo Blanco, Kevin Swersky, Arun Ahuja, Thi Avrahami, Vedant Misra, Raoul de Liedekerke, Mariko Iinuma, Alex Polozov, Sarah York, George van den Driessche, Paul Michel, Justin Chiu, Rory Blevins, Zach Gleicher, Adrià Recasens, Alban Rrustemi, Elena Gribovskaya, Aurko Roy, Wiktor Gworek, Sébastien M. R. Arnold, Lisa Lee, James Lee-Thorp, Marcello Maggioni, Enrique Piqueras, Kartikeya Badola, Sharad Vikram, Lucas Gonzalez, Anirudh Baddepudi, Evan Senter, Jacob Devlin, James Qin, Michael Azzam, Maja Trebacz, Martin Polacek, Kashyap Krishnakumar, Shuoyiin Chang, Matthew Tung, Ivo Penchev, Rishabh Joshi, Kate Olszewska, Carrie Muir, Mateo Wirth, Ale Jakse Hartman, Josh Newlan, Sheleem Kashem, Vijay Bolina, Elahe Dabir, Joost van Amersfoort, Zafarali Ahmed, James Cobon-Kerr, Aishwarya Kamath, Arnar Mar Hrafnkelsson, Le Hou, Ian Mackinnon, Alexandre Frechette, Eric Noland, Xiance Si, Emanuel Taropa, Dong Li, Phil Crone, Anmol Gulati, Sébastien Cevey, Jonas Adler, Ada Ma, David Silver, Simon Tokumine, Richard Powell, Stephan Lee, Kiran Vodrahalli, Samer Hassan, Diana Mincu, Antoine Yang, Nir Levine, Jenny Brennan, Mingqiu Wang, Sarah Hodkinson, Jeffrey Zhao, Josh Lipschultz, Aedan Pope, Michael B. Chang, Cheng Li, Laurent El Shafey, Michela Paganini, Sholto Douglas, Bernd Bohnet, Fabio Pardo, Seth Odoom, Mihaela Rosca, Cicero Nogueira dos Santos, Kedar Soparkar, Arthur Guez, Tom Hudson, Steven Hansen, Chulayuth Asawaroengchai, Ravi Addanki, Tianhe Yu, Wojciech Stokowiec, Mina Khan, Justin Gilmer, Jaehoon Lee, Carrie Grimes Bostock, Keran Rong, Jonathan Caton, Pedram Pejman, Filip Pavetic, Geoff Brown, Vivek Sharma, Mario Lučić, Rajkumar Samuel, Josip Djolonga, Amol Mandhane, Lars Lowe Sjösund, Elena Buchatskaya, Elspeth White, Natalie Clay, Jiepu Jiang, Hyeontaek Lim, Ross Hemsley, Zeyncep Cankara, Jane Labanowski, Nicola De Cao, David Steiner, Sayed Hadi Hashemi, Jacob Austin, Anita Gergely, Tim Blyth, Joe Stanton, Kaushik Shivakumar, Aditya Siddhant, Anders Andreassen, Carlos Araya, Nikhil Sethi, Rakesh Shivanna, Steven Hand, Ankur Bapna, Ali Khodaei, Antoine Miech, Garrett Tanzer, Andy Swing, Shantanu Thakoor, Lora Aroyo, Zhufeng Pan, Zachary Nado, Jakub Sygnowski, Stephanie Winkler, Dian Yu, Mohammad Saleh, Loren Maggiore, Yamini Bansal, Xavier Garcia, Mehran Kazemi, Piyush Patil, Ishita Dasgupta, Iain Barr, Minh Giang, Thais Kagohara, Ivo Danihelka, Amit Marathe, Vladimir Feinberg, Mohamed Elhawaty, Nimesh Ghelani, Dan Horgan, Helen Miller, Lexi Walker, Richard Tanburn, Mukarram Tariq, Disha Shrivastava, Fei Xia, Qingze Wang, Chung-Cheng Chiu, Zoe Ashwood, Khuslen Baatarsukh, Sina Samangooei, Raphaël Lopez Kaufman, Fred Alcober, Axel Stjerngren, Paul Komarek, Katerina Tsihlas, Anudhyan Boral, Ramona Comanescu, Jeremy Chen, Ruibo Liu, Chris Welty, Dawn Bloxwich, Charlie Chen, Yanhua Sun, Fangxiaoyu Feng, Matthew Mauger, Xerxes Dotiwalla, Vincent Hellendoorn, Michael Sharman, Ivy Zheng, Krishna Haridasan, Gabe Barth-Maron, Craig Swanson, Dominika Rogozińska, Alek Andreev, Paul Kishan Rubenstein, Ruoxin Sang, Dan Hurt, Gamaleldin Elsayed, Renshen Wang, Dave Lacey, Anastasija Ilić, Yao Zhao, Adam Iwanicki, Alejandro Lince, Alexander Chen, Christina Lyu, Carl Lebsack, Jordan Griffith, Meenu Gaba, Paramjit Sandhu, Phil Chen, Anna Koop, Ravi Rajwar, Soheil Hassas Yeganeh, Solomon Chang, Rui Zhu, Soroush Radpour, Elnaz Davoodi, Ving Ian Lei, Yang Xu, Daniel Toyama, Constant Segal, Martin Wicke, Hanzhao Lin, Anna Bulanova, Adrià Puigdomènech Badia, Nemanja Rakićević, Pablo Sprechmann, Angelos Filos, Shaobo Hou, Víctor Campos, Nora Kassner, Devendra Sachan, Meire Fortunato, Chimezie Iwuanyanwu, Vitaly Nikolaev, Balaji Lakshminarayanan, Sadegh Jazayeri, Mani Varadarajan, Chetan Tekur, Doug Fritz, Misha Khalman, David Reitter, Kingshuk Dasgupta, Shourya Sarcar, Tina Ornduff, Javier Snaider, Fantine Huot, Johnson Jia, Rupert Kemp, Nejc Trdin, Anitha Vijayakumar, Lucy Kim, Christof Angermueller, Li Lao, Tianqi Liu, Haibin Zhang, David Engel, Somer Greene, Anaïs White, Jessica Austin, Lilly Taylor, Shereen Ashraf, Dangyi Liu, Maria Georgaki, Irene Cai, Yana Kulizhskaya, Sonam Goenka, Brennan Saeta, Ying Xu, Christian Frank, Dario de Cesare, Brona Robenek, Harry Richardson, Mahmoud Alnahlawi, Christopher Yew, Priya Ponnapalli, Marco Tagliasacchi, Alex Korchemniy, Yelin Kim, Dinghua Li, Bill Rosgen, Kyle Levin, Jeremy Wiesner, Praseem Banzal, Praveen Srinivasan, Hongkun Yu, Çağlar Ünlü, David Reid, Zora Tung, Daniel Finchelstein, Ravin Kumar, Andre Elisseeff, Jin Huang, Ming Zhang, Ricardo Aguilar, Mai Giménez, Jiawei Xia, Olivier Dousse, Willi Gierke, Damion Yates, Komal Jalan, Lu Li, Eri Latorre-Chimoto, Duc Dung Nguyen, Ken Durden, Praveen Kallakuri, Yaxin Liu, Matthew Johnson, Tomy Tsai, Alice Talbert, Jasmine Liu, Alexander Neitz, Chen Elkind, Marco Selvi, Mimi Jasarevic, Livio Baldini Soares, Albert Cui, Pidong Wang, Alek Wenjiao Wang, Xinyu Ye, Krystal Kallarackal, Lucia Loher, Hoi Lam, Josef Broder, Dan Holtmann-Rice, Nina Martin, Bramandia Ramadhana, Mrinal Shukla, Sujoy Basu, Abhi Mohan, Nick Fernando, Noah Fiedel, Kim Paterson, Hui Li, Ankush Garg, Jane Park, DongHyun Choi, Diane Wu, Sankalp Singh, Zhishuai Zhang, Amir Globerson, Lily Yu, John Carpenter, Félix de Chaumont Quitry, Carey Radebaugh, Chu-Cheng Lin, Alex Tudor, Prakash Shroff, Drew Garmon, Dayou Du, Neera Vats, Han Lu, Shariq Iqbal, Alex Yakubovich, Nilesh Tripuraneni, James Manyika, Haroon Qureshi, Nan Hua, Christel Ngani, Maria Abi Raad, Hannah Forbes, Jeff Stanway, Mukund Sundararajan, Victor Ungureanu, Colton Bishop, Yunjie Li, Balaji Venkatraman, Bo Li, Chloe Thornton, Salvatore Scellato, Nishesh Gupta, Yicheng Wang, Ian Tenney, Xihui Wu, Ashish Shenoy, Gabriel Carvajal, Diana Gage Wright, Ben Bariach, Zhuyun Xiao, Peter Hawkins, Sid Dalmia, Clement Farabet, Pedro Valenzuela, Quan Yuan, Ananth Agarwal, Mia Chen, Wooyeol Kim, Brice Hulse, Nandita Dukkipati, Adam Paszke, Andrew Bolt, Kiam Choo, Jennifer Beattie, Jennifer Prendki, Harsha Vashisht, Rebeca Santamaria-Fernandez, Luis C. Cobo, Jarek Wilkiewicz, David Madras, Ali Elqursh, Grant Uy, Kevin Ramirez, Matt Harvey, Tyler Liechty, Heiga Zen, Jeff Seibert, Clara Huiyi Hu, Andrey Khorlin, Maigo Le, Asaf Aharoni, Megan Li, Lily Wang, Sandeep Kumar, Norman Casagrande, Jay Hoover, Dalia El Badawy, David Soergel, Denis Vnukov, Matt Miecnikowski, Jiri Simsa, Praveen Kumar, Thibault Sellam, Daniel Vlasic, Samira Daruki, Nir Shabat, John Zhang, Guolong Su, Jiageng Zhang, Jeremiah Liu, Yi Sun, Evan Palmer, Alireza Ghaffarkhah, Xi Xiong, Victor Cotruta, Michael Fink, Lucas Dixon, Ashwin Sreevatsa, Adrian Goedeckemeyer, Alek Dimitriev, Mohsen Jafari, Remi Crocker, Nicholas FitzGerald, Aviral Kumar, Sanjay Ghemawat, Ivan Philips, Frederick Liu, Yannie Liang, Rachel Sterneck, Alena Repina, Marcus Wu, Laura Knight, Marin Georgiev, Hyo Lee, Harry Askham, Abhishek Chakladar, Annie Louis, Carl Crous, Hardie Cate, Dessie Petrova, Michael Quinn, Denese Owusu-Afriyie, Achintya Singhal, Nan Wei, Solomon Kim, Damien Vincent, Milad Nasr, Christopher A. Choquette-Choo, Reiko Tojo, Shawn Lu, Diego de Las Casas, Yuchung Cheng, Tolga Bolukbasi, Katherine Lee, Saaber Fatehi, Rajagopal Ananthanarayanan, Miteyan Patel, Charbel Kaed, Jing Li, Shreyas Rammohan Belle, Zhe Chen, Jaclyn Konzelmann, Siim Põder, Roopal Garg, Vinod Koverkathu, Adam Brown, Chris Dyer, Rosanne Liu, Azade Nova, Jun Xu, Alanna Walton, Alicia Parrish, Mark Epstein, Sara McCarthy, Slav Petrov, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. ArXiv:2403.05530 [cs].

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. ArXiv:2307.09288 [cs].

Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. ZeroShotDataAug: Generating and Augmenting Training Data with ChatGPT. ArXiv:2304.14334 [cs].

Cynthia Van Hee, Els Lefever, and Veronique Hoste. 2018. SemEval-2018 Task 3: Irony Detection in English Tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana. Association for Computational Linguistics.

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*, Lecture Notes in Computer Science, pages 84–95, Cham. Springer International Publishing.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative Data Augmentation for Commonsense Reasoning. *arXiv:2004.11546 [cs]*. ArXiv: 2004.11546.

Akhila Yerukola, Mason Bretan, and Hongxia Jin. 2021. Data Augmentation for Voice-Assistant NLU using BERT-based Interchangeable Rephrase. *arXiv:2104.08268 [cs]*. ArXiv: 2104.08268.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. *arXiv:1804.09541 [cs]*. ArXiv: 1804.09541.

	Dataset	apparently reassembled from the cutting-room floor of any given daytime soap.
Negative	-P	It appears to have been pieced together from discarded scenes of any typical daytime soap opera.
	-ZS	The film was a major disappointment, lacking any coherent plot or engaging characters.
	-3S	It's a shame that such a promising concept was wasted on a poorly executed film.
	Dataset	béart and berling are both superb , while huppert ... is magnificent.
Positive	-P	Both Béart and Berling deliver exceptional performances, while Huppert... she is truly magnificent.
	-ZS	This film is a refreshing breath of fresh air, offering a unique and original take on familiar themes.
	-3S	This film is a delightful and heartwarming celebration of friendship, full of laughter and touching moments.

Table 6: Examples of generated sentences for the SST-2 dataset by GPT3.5.

Dawei Zhu, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, and Dietrich Klakow. 2023. Weaker Than You Think: A Critical Look at Weakly Supervised Learning. ArXiv:2305.17442 [cs].

A Generated sentences

B Prompts

In this section, we describe the prompts we used for ChatGPT and LLama2. For the paraphrasing strategy, we copy Piedboeuf and Langlais (2023), who simply asks for paraphrases in batch if the ratio is 1 and for multiple paraphrases if the ratio is larger than one. We refer to their paper for more detail on the prompting.

For the zero and three-shot generation, we use the template shown in Table 7.

We referred to the description given in the original papers of each dataset to craft informative prompts.

C Supplementary Results

DATASET_DESC	CLASS_DESC
movie reviews	negative or somewhat negative
	positive or somewhat positive
headline Fake/Real news classification	Real
	Fake
Ironic tweet detection	Non-Ironic Tweets
	Ironic Tweets
Ironic tweet detection	Tweets that are not ironic
	Tweets ironic by polarity contrast, where the polarity is inverted between the literal and intended evaluation
	Tweets ironic by Situational Irony, where a situation fails to meet some expectation
	Tweets ironic by Other type of Irony, where the Irony is neither by Polarity Contrast or by Situational Irony
Question classification	Questions about an abbreviation
	Questions about an entity (event, animal, language, etc)
	Question concerning a description (of something, a definition, a reason, etc)
	Questions about a human (description of someone, an individual, etc)
	Questions about a location
	Questions about something numerical (weight, price, any other number)

Table 7: Prompt patterns for the zero-shot strategies for ChatGPT and Llama. The prompt is of the form "Here are some examples of $CLASS_DESC from a dataset of $DATASET_DESC: $EXAMPLES. Can you create 10 more sentences of that class that would fit the dataset ".

	Baseline	Perfect	Copy	EDA	AEDA	BT	CBERT	CBART	T5	GPT3.5-P	GPT3.5-ZS	GPT3.5-3S	Llama2-P	Llama2-ZS	Llama2-3S
Baseline	0	0	0	20	10	10	30	20	20	0	40	30	20	30	30
Perfect	80	0	90	70	70	90	100	80	90	80	100	90	90	100	90
Copy	0	0	0	0	0	0	0	30	10	0	40	20	0	10	20
EDA	10	0	0	0	0	0	0	20	20	0	50	20	10	10	10
AEDA	10	0	0	10	0	10	20	20	20	20	40	30	20	20	30
BT	10	0	0	0	0	0	20	20	10	0	20	20	0	20	10
CBERT	0	0	0	0	0	0	0	10	10	0	30	20	10	0	10
CBART	0	0	0	0	0	0	10	0	0	0	20	10	0	0	0
T5	0	0	0	0	0	0	0	10	0	0	20	10	0	10	0
GPT3.5-P	30	0	20	20	10	10	30	40	50	0	60	30	0	40	20
GPT3.5-ZS	0	0	0	0	0	0	0	10	0	0	0	10	0	0	0
GPT3.5-3S	10	0	10	0	0	0	10	10	10	10	30	0	0	0	0
Llama2-P	0	0	0	0	0	0	10	10	0	0	30	20	0	10	10
Llama2-ZS	10	0	10	0	0	0	0	10	10	0	20	10	10	0	20
Llama2-3S	10	0	0	0	0	0	10	10	0	10	20	20	0	10	0

Figure 4: Percentage of times the row algorithm performs statistically better than the column algorithm, with a p-value threshold of 0.05 and using a two-tails paired t-test and with the starting sizes of 500/1000.

	SST2	Irony	IronyB	TREC6	SNIPS	Average
Baseline	87.7/88.8	67.0/68.1	43.9/45.9	84.5/87.9	95.6/96.2	75.8/77.4
Perfect	88.9*/89.7*	67.3/71.1*	46.0*/48.5*	87.6*/89.3	96.1*/96.9*	77.2/79.1
Copy	87.7/88.8	**66.2**/69.2	43.7/46.7	83.8/86.3	95.5/**96.4**	75.4/77.5
EDA	88.1/88.9	65.9/69.4	43.6/47.2*	82.5/86.1	95.8/96.3	75.2/77.6
AEDA	88.0/89.0	65.1/69.5	42.8/46.7	82.7/87.4	**96.1***/96.3	74.9/**77.8**
BT	88.0/89.2*	64.9/68.4	44.2/45.8	83.1/87.1	95.5/96.3	75.1/77.3
CBERT	87.6/88.6	64.8/**69.5**	43.6/46.1	82.9/84.7	95.4/96.3	74.9/77.0
CBART	87.7/88.7	66.0/68.8	43.6/45.1	81.2/82.5	95.9/96.2	74.9/76.3
T5	87.5/88.8	64.9/67.7	43.6/45.6	82.9/84.8	95.8/96.3	75.0/76.6
GPT3.5-P	**88.2***/**89.3***	65.1/**69.5***	**44.3/47.5**	**84.5/86.8**	95.5/96.1	**75.5/77.8**
GPT3.5-ZS	87.7/88.8	64.2/67.4	42.0/45.1	81.9/86.0	95.5/95.8	74.3/76.6
GPT3.5-3S	88.1*/88.6	64.9/69.0	40.4/44.4	83.0/86.8	95.9/96.2	74.5/77.0
Llama2-P	87.8/88.9	65.4/68.1	44.1/46.1	83.3/85.9	95.7/**96.4**	75.3/77.1
Llama2-ZS	88.1*/88.8	65.0/69.2	42.5/45.7	82.2/86.0	95.6/96.0	74.7/77.2
Llama2-3S	87.8/88.4	65.0/67.9	42.8/46.2	83.8/85.3	95.7/**96.4***	75.0/76.8

Table 9: Average metric over 15 runs for the training set sizes of 500 (left) and 1000 (right) with a ratio of 1. We report accuracy for binary tasks and macro-f1 for multiclass ones. STDs are between 0.6 and 3.0, depending on the dataset. Stars represent results for which the difference with the baseline was found to be statistically significant.