Domain-aligned Data Augmentation for Low-resource and Imbalanced Text Classification

Nikolaos Stylianou[0000−0002−6396−5374], Despoina Chatzakou[0000−0002−9564−7100], Theodora Tsikrika[0000−0003−4148−9028], Stefanos Vrochidis[0000−0002−2505−9178], and Ioannis Kompatsiaris[0000−0001−6447−9020]

Information Technologies Institute, Centre for Research and Technology Hellas
{nstylia,dchatzakou,theodora.tsikrika,stefanos,ikom}@iti.gr

Abstract. Data Augmentation approaches often use Language Models, pretrained on large quantities of unlabeled generic data, to conditionally generate examples. However, the generated data can be of subpar quality and struggle to maintain the same characteristics as the original dataset. To this end, we propose a Data Augmentation method for low-resource and imbalanced datasets, by aligning Language Models to in-domain data prior to generating synthetic examples. In particular, we propose the alignment of existing generic models in task-specific unlabeled data, in order to create better synthetic examples and boost performance in Text Classification tasks. We evaluate our approach on three diverse and well-known Language Models, four datasets, and two settings (i.e. imbalance and low-resource) in which Data Augmentation is usually deployed, and study the correlation between the amount of data required for alignment, model size, and its effects in downstream in-domain and out-of-domain tasks. Our results showcase that in-domain alignment helps create better examples and increase the performance in Text Classification. Furthermore, we find a positive connection between the number of training parameters in Language Models, the volume of fine-tuning data, and their effects in downstream tasks.

Keywords: Natural Language Processing · Data Augmentation · Low-Resource Data · Imbalanced Data · Text Classification

1 Introduction

Modern Deep Learning applications typically require a very large amount of labeled training data [30] to operate in a satisfactory manner. While some Large Language Models [39,6,29] have been capable of achieving state-of-the-art performance with only a handful of labeled examples, Few-Shot learning, in which a small number of examples are provided as contextualized prompts, remains a challenging task [5]. Data Augmentation acts as a countermeasure to the lack of sufficiently labeled training data, serving as an effective strategy in artificially increasing the size of a training dataset.

In addition, commonly, label distribution in the real world is rarely balanced, especially in the case of user generated content, a fact which is often reflected in the training data and thus resulting in poor performance of the trained models [14]. Solving class imbalance in the training data is usually dealt with over or under sampling techniques from the dominant or subservient class, respectively [8,25]. However, these approaches can lead to overfitting and loss of valuable information, respectively.

With Data Augmentation, the goal is to generate examples for specific classes such that the training dataset is increased. While traditional Natural Language Processing (NLP) Data Augmentation techniques [35,38] often struggle to maintain the correct label for the created examples, modern approaches utilizing Language Models (LMs) have made significant strides in this area [37,1,19]. Effectively, having the ability to automatically generate class-specific examples can significantly contribute to both low-resource scenarios where labeled data are scarce, as well as in tasks with label imbalance. What is more, as we avoid over and under sampling examples (e.g. by randomly creating copies or deleting examples from the minority and majority class, respectively) we minimize the risk of overfitting and information loss, respectively.

With LM-based Data Augmentation approaches, the use of LMs pretrained on generic domains often results in generated examples that do not reflect a task's specific characteristics. This paper explores the effects of in-domain alignment for LM-based Data Augmentation, i.e. fine-tuning pretrained LMs on domain specific data given a certain task. We evaluate our approach in both extremely low-resource and highly imbalanced settings for Text Classification. Specifically, we expand the work of [19], which establishes a method for utilizing popular and state-of-the-art LMs to create synthetic data.

Concretely, in this work, we fine-tune three diverse LMs (GPT2, BERT, and BART) on in-domain data to align the pretrained models with the domain's characteristics. This alignment is performed with three increasing sizes of in-domain data to examine the correlation between different LMs and in-domain data volume. We also investigate the effectiveness of Data Augmentation on datasets with user generated texts of differing lengths, namely the SST-2, SNIPS, and TREC datasets used in previous works [1,19] and also the Rotten Tomatoes (RT) dataset [26], which has significantly longer texts compared to the moderately sized texts of the other three datasets. Moreover, to further assess the robustness and validity of the in-domain alignment for Data Augmentation, we simulate extreme label imbalance for the in-domain corpora by under-sampling a single class of the training set. By artificially doubling and balancing the minority class example with domain-aligned LMs, we showcase the performance gains among aligned LM Data Augmentation, non-aligned LM Data Augmentation and no augmentation. Overall, we highlight the effects of in-domain alignment with a BERT-based classifier [12] for both in- and out-of-domain corpora.

To sum up, our contributions are as follows: (a) we showcase that in-domain alignment can help produce better results in low-resource and imbalanced Text Classification, (b) we investigate the volume of data required for alignment in

different models and settings, and (c) we show that minority class Data Augmentation following in-domain alignment can greatly improve the performance in imbalanced Text Classification tasks.

2 Related Work

Data Augmentation (DA) has taken many forms, from simple alterations to the initial content [35,13] to automatically generating artificial (synthetic) examples [1,19,27].

Early approaches to DA focus on artificially altering the text through a series of transformations. EDA [35] proposes the replacement of certain words, word swapping, and random insertion and deletion of words to alter the original content, while ADEA [17] adds artificial noise by inserting strings of punctuation marks. Similarly, [38] uses interpolation and n-gram smoothing as a means of introducing noise, while [32] proposes the use of back-translation, i.e. translating a sample to another language and converting it back to the original language. SMERTI [13] presents a semantic approach towards text replacement with the focus being on maintaining the sentiment and fluency of the original text.

Utilizing the advantages of LM pretraining, a plethora of approaches have been designed based on LM capabilities [4]. In general, LM-based DA approaches exploit the training objectives and text prompting techniques to condition the predictions of different LMs so that they can create artificial examples. Towards this direction, CBERT [37] exploits the Masked Language Modeling (MLM) objective of BERT [12] and fine-tunes the model so that the masked tokens are conditioned by the provided label. LAMBADA [1] utilizes GPT2 [28], a generative LM, to create labeled examples after fine-tuning it on the training set via label prompting. These examples are first filtered by a baseline classifier to ensure that the predicted labels correspond to the one used to generate them, before being used in the final training set. Similarly, a per-instance prompted GPT2 model is presented in [3], to create similar texts which are then filtered based on embedding distances.

Based on these approaches, [19] proposes a unified framework in which different conditioning strategies are explored for BERT, GPT2, and BART [20] in a low-resource scenario. In comparison to past approaches, no filtering is performed due to the extremely low number of samples available for training. To counteract the disadvantages of the early approaches, such as being task-specific or having difficulty to create label-preserving examples, Polyjuice [36] aims to generate texts with specific perturbations and substitutions. This is achieved through a counterfactual sentence generation process rather than label conditioning used in all other LM-based methods.

Several methods have also been proposed to augment the data in latent space, rather than creating new examples. Specifically, Cutoff [31] creates noise in the latent space by zeroing rows or columns of the input vectors, hence removing tokens, features, or even spans of words, without the need to artificially create examples. Building upon this, HiddenCut [9] uses Cutoff layers within the

Transformer architecture blocks [34] of the entire model to introduce noise between layers, similarly to how dropout works [2]. Similarly, Mixup [15] proposes a linear intepolation of textual samples from the same class to increase the input signals received by the model with the same number of available data. Lastly, CoDa [27] combines classic approaches, such as back-translation, with novel approaches, such as Cutoff and Mixup, as well as adversarial training to create better models. A recent survey on DA methods for text classification analyzes the aforementioned approaches in detail [3].

Previous approaches either fine-tune the respective LMs on the training data or generate latent examples which are hard to benchmark and tied to datasets. Consequently, we opt to focus on methods which can be evaluated with more means than only through the performance changes in downstream tasks. In particular, prior to fine-tuning on the training data, we first fine-tune the LMs on unlabeled in-domain data, to align the general language models with our effective domain such that the artificially generated examples better match the domain's characteristics and can therefore lead to better overall performance.

3 Domain-aligned Data Augmentation

Our proposed method aims to utilize well-established pretrained LMs and a collection of task-specific unlabeled data, to generate synthetic examples in order to increase dataset size and eliminate imbalance. By initially fine-tuning the pretrained LMs on in-domain data, we aim to generate better examples for the task in hand and increase performance.

To that end, we build on top of [19] that uses three language models, BERT, GPT2, and BART, which are autoencoding (AE), autoregressive (AR) and sequence-to-sequence (seq2seq) LMs, respectively. These models are aligned with the domain-specific data following their original training objectives (Section 3.1), while the conditional generation is informed by the findings of [19] (Section 3.2).

Problem formulation. Let $D_{Train} = \{x_i, y_i\}_n^1$ be a dataset for a task T containing n examples, where $x_i = \{w_j\}_m^1$ is an example in the dataset containing m words and y_i is the associated label of this example. And let $D_{Domain} = \{x_k\}_v^1$ be a dataset of v unlabeled examples, which can be easily acquired to match with task T. We assume that $v >> n$ with v being able to scale by acquiring more unlabeled available data. Also, let G be a LM pretrained on generic data. This work proposes $G_{Aligned}$ being the G fine-tuned on D_{Domain} and $G_{AlignedTuned}$ being the fine-tuned $G_{Aligned}$ LM on D_{Train}, while previous work has focused on G_{Tuned}, i.e., pretrained LMs fine-tuned only on D_{Train}. Finally, $D_{Synthetic}$ is the product of generating and adding s examples to D_{Train} using $G_{AlignedTuned}$ from a dataset $D_{Select} \subseteq D_{Train}$ containing only class examples under a threshold d from D_{Train}.

Our goal is to train a task specific model M such that $\texttt{Score}(M_{Synthetic}) > \texttt{Score}(M_{Train})$ trained on $D_{Synthetic}$ and D_{Train} respectively and `Score` is a task appropriate metric (e.g. Accuracy, F1-score, etc.). The process followed to achieve this is described in Algorithm 1.

Algorithm 1 Domain Aligned Data Augmentation

Input: Training dataset D_{Train}
In-domain unlabeled data D_{Domain}
Pretrained model G
d: threshold number of examples per class
s: number of examples to be generated

Fine-tune G using D_{Domain} to obtain $G_{Aligned}$
Fine-tune $G_{Aligned}$ using D_{Train} to obtain $G_{AlignedTuned}$
$D_{Synthetic} \leftarrow \{\}$
$D_{Select} \leftarrow \texttt{ExampleSelector}(D_{Train}, d)$
foreach $\{x_i, y_i\} \in D_{Select}$ **do**
    Synthesize s examples $\{\hat{x}_i, \hat{y}_i\}_s^1$ using $G_{AlignedTuned}$
    $D_{Synthetic} \leftarrow D_{Synthetic} \cup \{\hat{x}_i, \hat{y}_i\}_s^1$
end foreach
$D_{Task} \leftarrow D_{Train} \cup D_{Synthetic}$

3.1 In-Domain Alignment

For the LM alignment, we fine-tune the pretrained LMs on D_{Domain} using different training objectives to obtain $G_{Aligned}$. Specifically, we tune BERT using only the MLM objective and discard the Next Sentence Prediction objective described in [12], as it has shown to not contribute towards better performance in downstream tasks [22]. GPT2 is tuned following the original objective, in an autoregressive setting [28]. Lastly, BART uses the denoising objective described in [20], following the same corruption strategies. For brevity, we do not describe these objectives and methods in detail and refer readers to the cited works.

3.2 Conditional Generation

Using the previously acquired $G_{Aligned}$, we further fine-tune the LMs on D_{Select} of each task so that we obtain $G_{AlignedTuned}$ which will be capable to conditionally generate new instances. Fine-tuning strategies for conditional generation are also different for each individual model. For BERT, we follow the CBERT approach [37], in which the model is first fine-tuned with a MLM objective with the class as a single token sentence followed by a separator token and the original text. After the model is tuned, random tokens from the original text are masked and the model predicts replacements, altering the original text. For GPT2, we follow [1] in which the class is also prepended to the original text followed by a separator token and the model is trained autoregressively. Lastly, BART operates similarly to CBERT with the label prepended to the start of the original text, followed by a separator token [19]. For BART, we present results in two masking strategies, word and span, due to performance variance based on masking [19].

Readers are encouraged to follow the original works for details on the conditional generation process for each LM architecture. Conditional Generation is dynamically used for both low-resource and imbalanced data through the example selection threshold d. In the case of low-resource data, all labels are used

simultaneously to obtain $G_{AlignedTuned}$ (i.e. $D_{Select} = D_{Train}$), while for imbalanced data the model is trained only on the minority classes samples and produces only examples for those classes (i.e. $D_{Select} \subseteq D_{Train}$).

3.3 Baseline Classifier

The task-specific model (M) is a BERT-based classifier, with a dropout layer and a feed-forward layer with Softmax activation. Specifically, for each input sequence, the latent representation of the *[CLS]* special token, which acts as a sentence representation, is forwarded through the added layers to get the final label prediction. We opt to use this as our baseline to closely match with previous works [37,1,19]. All results presented in Section 5 are the effects of training this model, with the same configuration, on different datasets.

4 Experimental Setting

4.1 Datasets

We evaluate the proposed approach using four classification datasets, two of which belong to the same domain, namely Movie Reviews are examined in this work, and two are out-of-domain, along with a single in-domain dataset which we consider as unlabeled for domain alignment. The datasets used are:

- **SST-2** [33] a binary sentiment classification dataset (positive and negative) on movie reviews.
- **RT** [26] a binary sentiment classification dataset (positive and negative) on long movie reviews.
- **SNIPS** [11] an intent classification dataset, identifying seven distinct intents (PlayMusic, GetWeather, RateBook, SearchScreeningEvent, SearchCreativeWork, AddToPlaylist, BookRestaurant).
- **TREC** [21] a question classification dataset identifying six question types (Description, Entity, Abbreviation, Human, Location, Numeric).

Taking into account the classification datasets chosen for our experiments, we use the **IMDB** dataset [24], i.e. a binary sentiment (positive and negative) classification dataset on movie reviews, as a Domain alignment corpus. As a result, we consider SST-2 and RT as in-domain datasets and SNIPS and TREC as out-of-domain datasets. We opted for standardized and simple datasets for our experiments so that they are easy to replicate and focused on our objective, rather than introducing multiple levels of complexity. Similarly, for SST-2, SNIPS and TREC we use the same dataset versions as previous works [37,19], while for RT we use the published version of the dataset.

Detailed statistics of the datasets used in this work are presented in Table 1. Importantly, we note that all the datasets used in previous studies (i.e., SST-2, SNIPS, and TREC) contain small examples, with a maximum of 52 words per example across all datasets, while the RT and IMDB datasets we introduce contain

Data	No. Examples (train/dev/test)	Example Length (max/min/mean)
SST-2	6228/692/1821	52/2/19.28
RT	1200/400/400	2737/17/765.75
SNIPS	13084/700/700	35/2/9.09
TREC	4906/546/500	37/3/10.21
IMDB	75000/0/25000	2470/10/233.77

Table 1. Data statistics for datasets used in downstream tasks and alignment.

Dataset		10%	30%	50%
SST-2	POS	324	972	1622
	NEG	298	896	1494
RT	POS	62	184	310
	NEG	67	189	315

Table 2. Number of instances of the minority class after simulated imbalance.

longer sequences, with RT being used for evaluation and IMDB for in-domain alignment. Finally, the IMDB training set contains 50000 unlabeled examples, with the number of labeled examples being equal between training and testing.

4.2 In-Domain Data Volumes

LM fine-tuning aims to align the model while avoiding catastrophic forgetting. To fine-tune the three diverse LMs, the volume of data required and its effects in downstream tasks varies. As such, we split the IMDB train set in three volumes, *Small*, *Medium*, and *Large* of 18750, 37500, and 75000 examples, respectively, representing the 25%, 50%, and 100% of the training data. The volumes contain equal parts of positive, negative, and unlabeled examples.

4.3 Low resource and Imbalance Experiments

Simulating low resource. To align with previous work, we perform experiments on both in-domain and out-of-domain datasets. In particular, similar to [19], we create folds containing 10 examples of each class for all datasets. Specifically, for SST-2, SNIPS, and TREC we create 15 folds by randomly sampling examples, while for RT we use the provided 10 folds of the original dataset and sub-sample the examples in those. For each fold, we double the training size by creating 1 synthetic example ($s = 1$) for every original example (Algorithm 1).

Simulating imbalance. To evaluate the effects of our approach in different imbalanced scenarios, we perform experiments only on the in-domain datasets, given that this work focuses on in-domain alignment. In particular, to investigate the effects on various degrees of imbalance, we create three imbalanced datasets per class by sub-sampling each class of the training set for SST-2 and RT in 10%, 30% and 50% of all the class examples, while keeping the full number of examples of the opposite class. For each sub-sampling class, we create 5 folds through random example selection, to account with variance in the results. We evaluate both doubling the minority class examples ($s = 1$) as well as balancing the datasets ($s = 9$ for 10%, $s = 3$ for 30% and $s = 1$ for 50%), for each original minority example. Table 2 reports the number of instances of the minority class for each imbalance scenario.

4.4 Experimental Setup

We use three pretrained LMs for in-domain alignment and conditional generation. Their configuration during in-domain alignment is described in Table 3. All models were trained until convergence to the *Small*, *Medium*, and *Large* volumes of in-domain data.

Parameters	GPT2	BERT	BART
Common Name	"base"	"base-uncased"	"base"
Training Parameters	117M	110M	140M
Hidden Size	768	768	768
No Encoder/Decoder layers	12/0	0/12	6/6
No Attention Heads	12	12	12
Learning Rate	5e-6	5e-6	5e-6
Batch Size	64	64	32

Table 3. Language Model configuration for In-Domain Alignment.

For Conditional Generation in both Low-resource and Imbalance Setting, the models use the following configurations. GPT2 is trained for 25 epochs with a batch size of 32 and learning rate of 4e-5. Nucleus sampling is used for text generation with `top_p` 0.9, `top_k` 0 and `temperature` 0 [16]. BERT is trained for 10 epochs with a batch size of 8 and a learning rate of 4e-5. The masking probability is set to 15% and the maximum number of masked tokens is 256 per sequence. BART is trained for 30 epochs with a batch size of 12 and a learning rate of 1e-5. Token masking rate in both word and span masking is 40%. The number of examples generated depends on the experiment setting (Section 4.3) and the threshold (d) is set so that it selects all classes in low-resource setting and only the minority class in imbalanced setting.

Lastly, our Baseline classifier is using the `bert-base-uncased` configuration (Table 3), trained for 8 epochs with a batch size of 8 and a learning rate of 4e-5. In all experiments we use the Adam optimizer [18]. The codebase to reproduce all the experiments is available on Github[1].

5 Results

We consider two sets of results, for *low-resource setting* and for *imbalance setting*. In each setting, we compare our baseline classifier's performance trained on data with no augmentation, augmented using pretrained LMs (*Tuned*) [19], and augmented using domain-aligned LMs (*AlignedTuned*); the latter two double the minority class size, i.e., consider $s = 1$. In the imbalance setting, *Balanced* is further used to denote models which generate enough examples to balance

[1] https://github.com/M4D-MKLab-ITI/Domain-aligned-Data-Augmentation

the class instances, namely 9, 3, and 1 examples per minority class example, for 10%, 30%, and 50% imbalance, respectively (see also Section 4.3).

All results presented are product of re-implementation (including the results from the *Tuned* methods) due to the degree of randomness in the example selection process for the fold generation of the SST2, SNIPS, and TREC datasets.

Low-resource setting. In this setting, Table 4 presents the results with no augmentation, the results with augmentation using pretrained LMs (*Tuned*), and the results with augmentation using domain-aligned LMs (*AlignedTuned*) by first listing per LM the best result among the three volume sizes, followed by the results per LM for each volume size.

Model	**SST-2**	**RT**	**TREC**	**SNIPS**
NoAug	52.817±4.174	55.5±6.5	43.506±11.364	85.714±2.794
GPT2$_{\text{Tuned}}$	57.250±5.998	57.5±7.158	55.146±9.912	85.714±2.794
CBERT$_{\text{Tuned}}$	59.549±5.706	61.5±9.233	57.146±8.554	87.171±3.452
BARTword$_{\text{Tuned}}$	59.205±4.168	60.5±7.566	58.786±6.193	85.476±3.198
BARTspan$_{\text{Tuned}}$	59.769±3.976	61.5±7.762	57.386±8.599	87.074±2.835
GPT2$_{\text{AlignedTuned}}$	**58.290**±5.071	**59.0**±6.244	**57.426**±4.768	**87.019**±3.163
CBERT$_{\text{AlignedTuned}}$	**60.505**±6.137	**65.5**±7.889	**59.986**±8.107	**87.847**±2.128
BARTword$_{\text{AlignedTuned}}$	**60.464**±4.628	**64.0**±12.806	56.040±8.208	**85.876**±2.719
BARTspan$_{\text{AlignedTuned}}$	58.528±4.198	**65.0**±7.416	55.200±8.304	85.477±2.680
Small Volume				
GPT2$_{\text{AlignedTuned}}$	57.744±5.262	58.0±6.403	54.666±8.514	85.809±2.885
CBERT$_{\text{AlignedTuned}}$	**60.505**±6.137	63.0±9.797	**59.986**±8.107	**87.847**±2.128
BARTword$_{\text{AlignedTuned}}$	58.586±5.085	**64.0**±12.806	56.040±8.208	**85.876**±3.676
BARTspan$_{\text{AlignedTuned}}$	58.528±4.198	64.0±9.695	55.200±8.304	85.477±2.680
Medium Volume				
GPT2$_{\text{AlignedTuned}}$	**58.290**±5.071	**59.0**±6.244	55.133±7.838	86.828 ±2.872
CBERT$_{\text{AlignedTuned}}$	60.505±6.137	63.0±9.797	59.986±8.107	87.847±2.128
BARTword$_{\text{AlignedTuned}}$	**60.464**±4.628	61.5±11.191	55.093±10.711	85.847±2.719
BARTspan$_{\text{AlignedTuned}}$	58.455±4.493	63.5±7.762	50.386±10.832	84.057±3.553
Large Volume				
GPT2$_{\text{AlignedTuned}}$	57.426±4.768	59.0±8.306	**57.426**±4.768	**87.019**±3.163
CBERT$_{\text{AlignedTuned}}$	58.056±5.765	**65.5**±7.889	58.866±8.161	87.647±2.360
BARTword$_{\text{AlignedTuned}}$	60.464±4.628	61.5±11.191	53.306±10.421	84.561±3.465
BARTspan$_{\text{AlignedTuned}}$	58.191±4.034	**65.0** ±7.416	49.933±8.806	85.028±2.585

Table 4. Low-resource classifier performance with in-domain aligned models. Bold scores are the best score *per model per dataset*. Underlined scores indicate improvement over lower volume alignment and non-augmented scores *per model per dataset*. Bold & underlined scores are best scores *per dataset*.

For in-domain evaluation, we notice that, on SST-2 and RT, the classifier achieves overall better performance when trained on examples created from domain-aligned LMs. Comparing the best performing aligned model with their non-aligned counterpart, statistical significant performance increase ($p < 0.05$)

is achieved in all experiments on the RT dataset[2] and we see a noticeable but not significant increase on SST-2 ($p \approx 0.20$). As an exception, the aligned BART with span masking on the SST-2 dataset exhibits a loss in overall Accuracy. This comes in line with the findings of [19], where different BART masking strategies perform better in different datasets. In out-of-domain evaluation, on TREC and SNIPS, the results reveal increase in Accuracy, but no statistically important, when the examples are generated from GPT2 and CBERT, while aligned BART generated examples generally appear to have an opposite effect.

Examining the performance of the domain-aligned LMs further, with respect to the volume sizes, we notice that in-domain performance tends to increase with volume size, before it slightly drops. However, even the degraded scores are overall better than the ones achieved by the non-aligned LM generated examples. In out-of-domain datasets, we notice a performance drop, proportional to the level of alignment of the LMs with the task in hand; exceptions rise in the form of CBERT and GPT2 where we notice an increase in mean Accuracy when aligned on *Small* and *Large* volumes respectively.

Imbalance setting. For imbalanced evaluation, we test three different levels of imbalance with F1-score using, for the synthetic data generation, the best aligned LMs in terms of data volume from each LM type (see Table 4 for the best performing *AlignedTuned* LMs). We opt to use F1-score, as Accuracy is plagued with majority class bias and hence inherently flawed in this setting [7,23]. We compare the classifier's performance with examples generated from the aligned LMs to that with examples generated from pretrained LMs, as well as without any augmentation.

SST-2						
Model	Pos 10%	Neg 10%	Pos 30%	Neg 30%	Pos 50%	Neg 50%
NoAug	82.65±0	33.30±0	89.24±0	86.63±0	90.14±0	89.97±0
$\text{GPT2}_{\text{Tuned}}$	77.64±3.5	56.41±12.1	88.06±1.0	83.06±1.9	89.36±1.1	87.61±5.0
$\text{GPT2}_{\text{AlignedTuned}}$	**78.81**±2.8*	**58.63**±5.1*	**88.56**±0.5	**83.53**±1.6	**89.57**±0.5	**88.37**±0.2*
$\text{GPT2}_{\text{TunedBalanced}}$	57.82±1.0	54.72±1.3	75.13±2.4	72.58±2.2	89.36±1.1	87.61±5.0
$\text{GPT2}_{\text{AlignedTunedBalanced}}$	58.24±1.3	55.03±1.1	76.26±2.1*	73.31±1.9	**89.57**±0.5	**88.37**±0.2*
$\text{CBERT}_{\text{Tuned}}$	87.48±0.6	77.23±1.6	90.5±0.6	88.32±0.7	91.03±0.2	90.29±0.6
$\text{CBERT}_{\text{AlignedTuned}}$	**87.99**±0.7	**78.64**±0.9*	**90.52**±0.2	**88.72**±0.7*	**91.4**±0.2	**90.58**±0.4*
$\text{CBERT}_{\text{TunedBalanced}}$	84.64±1.2	76.34±2.7	90.11±0.4	87.68±0.6	91.03±0.2	90.29±0.6
$\text{CBERT}_{\text{AlignedTunedBalanced}}$	85.12±0.6	77.95±1.2*	90.26±0.4	88.70±0.3*	**91.4**±0.2	**90.58**±0.4*
$\text{BART}_{\text{Tuned}}$	82.68±3.0	77.63±2.4	89.79±0.5	86.77±1.2	90.79±0.4	89.22±0.4
$\text{BART}_{\text{AlignedTuned}}$	**84.56**±1.4*	**78.10**±1.3*	**90.10**±0.3*	87.11±0.1*	**91.12**±0.3	**89.47**±0.5
$\text{BART}_{\text{TunedBalanced}}$	83.5±0.7	77.14±1.2	89.78±0.8	88.40±0.7	90.79±0.4	89.22±0.4
$\text{BART}_{\text{AlignedTunedBalanced}}$	84.32±1.0	77.70±1.0	89.88±0.2	**88.60**±0.6	**91.12**±0.3	**89.47**±0.5

Table 5. F1-score score of imbalanced setting with artificial imbalance in Positive (Pos) or Negative (Neg) label for the SST-2 dataset. Bold scores are the best score *per model per dataset*. Bold & underlined scores are best scores per dataset. Statistical significant improvement ($p < 0.05$) over non-alinged counterparts is shown with *.

[2] $\text{BARTword}_{\text{Aligned}}$ has a $p < 0.06$ due to high STD.

Starting with SST-2 (Table 5), we note that the aligned models performed better than their non-aligned counterparts in all settings, with better F1-score and lower standard deviation among folds. However, not all improvements are statistically important, with all GPT2 models failing to improve over baseline results. In addition, performance tends to degrade when artificially balancing the instances with all models.

Interestingly, statistical significant improvements are mostly noted when the Negative class is the subservient one, which maintains lower scores in all settings than with a Positive subservient class. These results verify an inherit difficulty in predicting negative examples due to high average mutual information between class examples, pointing to high similarity between same class texts, the use of sarcasm which portraits them as positive texts, and noise due to mislabeled examples [10].

RT						
Model	Pos 10%	Neg 10%	Pos 30%	Neg 30%	Pos 50%	Neg 50%
NoAug	33.33±0	33.33±0	73.29±2.2	33.33±0	83.85±0	64.06±0
$GPT2_{Tuned}$	33.33±0	33.33±0	34.52±1.8	39.67±14.5	80.51±3.6	71.21±16.23
$GPT2_{AlignedTuned}$	33.33±0	33.33±0	42.00±8.3*	45.12±12.6*	**81.31**±5.2	**78.56**±6.5*
$GPT2_{TunedBalanced}$	59.95±5.6	62.10±5.4	80.06±2.9	70.03±13.8	80.51±3.6	71.21±16.23
$GPT2_{AlignedTunedBalanced}$	**61.21**±11.1*	**63.25**±3.5*	**81.11**±1.1	**78.09**±3.7*	**81.31**±5.2	**78.56**±6.5*
$CBERT_{Tuned}$	33.33±0	33.33±0	70.74±20.0	57.04±20.9	72.17±19.5	76.22±19.4
$CBERT_{AlignedTuned}$	33.33±0	33.33±0	75.60±11.8*	58.92±20.9*	**84.41**±0.9*	**82.35**±3.5*
$CBERT_{TunedBalanced}$	74.92±4.5	65.59±6.4	82.39±2.2	73.39±3.3	72.17±19.5	76.22±19.4
$CBERT_{AlignedTunedBalanced}$	**76.19**±3.0*	**68.69**±4.2*	**84.18**±1.2*	**74.38**±2.9*	**84.41**±0.9*	**82.35**±3.5*
$BART_{Tuned}$	33.33±0	33.33±0	39.28±11.9	50.38±21.1	37.38±20.0	83.37±2.5
$BART_{AlignedTuned}$	33.33±0	33.33±0	39.50±11.8	57.47±18.8*	**83.39**±0.5*	**84.47**±2.4*
$BART_{TunedBalanced}$	61.82±13.0	55.69±6.9	80.60±3.2	73.84±20.3	37.38±20.0	83.37±2.5
$BART_{AlignedTunedBalanced}$	**62.32**±9.5*	**66.11**±6.2*	**81.72**±3.3	**81.74**±4.4*	**83.39**±0.5*	**84.47**±2.4*

Table 6. F1-score of imbalanced setting with artificial imbalance in Positive (Pos) or Negative (Neg) label for the RT dataset. Bold scores are the best score *per model per dataset*. Bold & underlined scores are best scores per dataset. Statistical significant improvement ($p < 0.05$) over non-alinged counterparts is shown with *.

In RT (Table 6), the performance advantage between Positive and Negative subservient class datasets is more equally divided, depending on the augmentation approach. In addition, in this longer text dataset the alignment gains are more pronounced with almost all aligned models achieving statistically significant improvement over their non aligned counterparts. More importantly, when the subservient class becomes balanced, we notice dramatic improvement in performance compared to just doubling the minority class examples.

Specifically, we note that in all the 10% imbalance cases on RT, the augmentation proved ineffective when doubling the subservient class examples and the classifier failed to predict the minority class. In comparison, the balanced counterparts almost double the performance in the 10% and 30% imbalance set-

tings. This jump in performance is attributed both to the extra examples and the quality of the generated examples, which are longer and allow for more diversity.

Overall, balancing the minority class showcases very different behavior in SST-2 and RT, with the first degrading and the second improving. This is attributed to both the size of the datasets, where the subservient class at 50% imbalance of RT has the same number of instances as the subservient class at 10% imbalance of SST-2, and their characteristics, i.e., short and long texts that grant different levels of generation freedom to the models. We intrinsically notice less diverse generated examples for SST-2 balancing, which led to overfitting issues of the characterstics of the generated examples. In RT, this phenomenon is universally less pronounced, leading to the increase in performance.

6 Discussion

Overall, we show that in-domain alignment can have a positive effect in example generation when the labeled set of training data for a certain classification task is either very small or characterized by imbalance.

In particular, we observe that in low-resource settings, using only a small amount of labeled data, we can generate synthetic examples for all classes, boosting the performance of the text classifier. Furthermore, by only creating synthetic examples of the minority classes in imbalanced scenarios, we significantly help improve the performance, especially in the case of RT. Importantly, the experimental results showcase even when balancing the dataset, synthetic examples are not a replacement for real examples, but they can lead to significantly better performance. We further found that generating more than one synthetic example from each training example, longer sequences allowed for higher degree of freedom to the model, which translated into improved performance. Short texts on the other hand exhibited the opposite effect, with repeated synthetic examples.

In addition, by studying the effects of unlabeled data volume in the downstream tasks, we notice that different LM architectures operate best under different data volumes. A correlation between the number of training parameters of the LM and the volume of data exists, with CBERT operating better in the *Small* and *Medium* ranges, while also having the best lowest parameter count, GPT2 performing best in the *Medium* range, and BART performing best in the *Medium* and *Large* ranges.

This fluctuation in volume sizes of the same LM architecture is attributed to the different characteristics of the in-domain datasets. RT is characterized by considerably larger sequences than SST-2 and we dynamically select the number of tokens to be replaced or generated depending on the architecture. Hence, the longer the sequence, the more predictions are required by the LM and the harder it becomes to generate quality examples.

In terms of out-of-domain evaluation, performance generally degrades proportionally to the level of LM alignment. While this is expected, it is important to highlight it, as it limits the usability of the aligned LMs in other tasks. Overall, GPT2 appears to handle best out-of-domain tasks, which can be attributed

to its autoregressive nature, while both other models replace only parts of the original examples through masking predictions.

However, in imbalanced settings and especially on the RT dataset we find that model performance varies depending on the class examples to be generated and the quantity of available training data. With positive examples as the minority class, performance is overall better in all models, with the exception of GPT2 which excels in generating negative examples on the RT dataset. As the same behavior is exhibited in both aligned and non-aligned models, it cannot be attributed to the alignment dataset, but can be attributed to the models' characteristics.

Overall, the aligned models improved over their non-aligned counter parts in both low-resource and imbalanced settings with mask-based augmentation models (i.e, BERT and BART) performed better than their generative counterpart (i.e., GPT2). However, examples generated from GPT2 were more diverse and different generation strategies can significantly impact performance.

7 Conclusions and Future Work

Current Data Augmentation approaches focus on either very specific augmentation techniques which are hard to transfer to other tasks or generic approaches to filter large quantities of automatically created examples. We propose the use of in-domain alignment for LM-based Data Augmentation in low-resource and imbalance Text Classification tasks.

By aligning the model, better synthetic examples are generated that can boost the performance of the in-domain tasks. We also find a positive correlation between volume of unlabeled data for in-domain alignment and downstream performance, as well as identify performance degrading point which can inform future applications.

While our approach creates better examples, generating a plethora of examples from a single example is non-trivial, as evident by our experimental results when balancing imbalanced datasets. Improving the text generation process so that it better scales to the generation of more examples, while remaining invariant to text characteristics, remains a future work. Besides the creation of better examples from LMs, our approach can be bootstrapped to semi-supervised or active learning approaches, following other works, that can help filter out generated examples and increase the performance further.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 101021797 (STARLIGHT) and No 833464 (CREST).

References

1. Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., Tepper, N., Zwerdling, N.: Do not have enough data? deep learning to the rescue!

In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 7383–7390 (2020)
2. Baldi, P., Sadowski, P.J.: Understanding dropout. Advances in neural information processing systems **26** (2013)
3. Bayer, M., Kaufhold, M.A., Buchhold, B., Keller, M., Dallmeyer, J., Reuter, C.: Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. International Journal of Machine Learning and Cybernetics pp. 1–16 (2022)
4. Bayer, M., Kaufhold, M.A., Reuter, C.: A survey on data augmentation for text classification. ACM Computing Surveys **55**(7), 1–39 (2022)
5. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
6. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
7. Brzezinski, D., Stefanowski, J., Susmaga, R., Szczech, I.: On the dynamics of classification measures for imbalanced and streaming data. IEEE transactions on neural networks and learning systems **31**(8), 2868–2878 (2019)
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
9. Chen, J., Shen, D., Chen, W., Yang, D.: HiddenCut: Simple data augmentation for natural language understanding with better generalizability. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4380–4390. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.338
10. Collins, E., Rozanov, N., Zhang, B.: Evolutionary data measures: Understanding the difficulty of text classification tasks. In: Proceedings of the 22nd Conference on Computational Natural Language Learning. pp. 380–391 (2018)
11. Coucke, A., Saade, A., Ball, A., Bluche, T., Caulier, A., Leroy, D., Doumouro, C., Gisselbrecht, T., Caltagirone, F., Lavril, T., et al.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint arXiv:1805.10190 (2018)
12. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186 (2019)
13. Feng, S.Y., Li, A.W., Hoey, J.: Keep calm and switch on! preserving sentiment and fluency in semantic text exchange. 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). pp. 2701–2711. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1272
14. Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F.: Learning from imbalanced data sets, vol. 10. Springer (2018)
15. Guo, H., Mao, Y., Zhang, R.: Augmenting data with mixup for sentence classification: An empirical study. arXiv preprint arXiv:1905.08941 (2019)

16. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. In: International Conference on Learning Representations (2020)
17. Karimi, A., Rossi, L., Prati, A.: AEDA: An easier data augmentation technique for text classification. In: Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 2748–2754. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Kumar, V., Choudhary, A., Cho, E.: Data augmentation using pre-trained transformer models. In: Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems. pp. 18–26. Association for Computational Linguistics, Suzhou, China (Dec 2020)
20. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880 (2020)
21. Li, X., Roth, D.: Learning question classifiers. In: COLING 2002: The 19th International Conference on Computational Linguistics (2002)
22. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
23. Luque, A., Carrasco, A., Martín, A., de Las Heras, A.: The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition **91**, 216–231 (2019)
24. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 142–150. Association for Computational Linguistics, Portland, Oregon, USA (June 2011)
25. Mani, I., Zhang, I.: knn approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of workshop on learning from imbalanced datasets. vol. 126, pp. 1–7. ICML (2003)
26. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04). pp. 271–278 (2004)
27. Qu, Y., Shen, D., Shen, Y., Sajeev, S., Chen, W., Han, J.: Coda: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding. In: International Conference on Learning Representations (2020)
28. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
29. Rae, J.W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., et al.: Scaling language models: Methods, analysis & insights from training gopher. arXiv preprint arXiv:2112.11446 (2021)
30. Rosenfeld, J.S.: Scaling laws for deep learning. arXiv preprint arXiv:2108.07686 (2021)
31. Shen, D., Zheng, M., Shen, Y., Qu, Y., Chen, W.: A simple but tough-to-beat data augmentation approach for natural language understanding and generation. arXiv preprint arXiv:2009.13818 (2020)

32. Shleifer, S.: Low resource text classification with ulmfit and backtranslation. arXiv preprint arXiv:1903.09244 (2019)
33. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing. pp. 1631–1642 (2013)
34. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
35. Wei, J., Zou, K.: 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). pp. 6382–6388. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1670
36. Wu, T., Ribeiro, M.T., Heer, J., Weld, D.S.: Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 6707–6723 (2021)
37. Wu, X., Lv, S., Zang, L., Han, J., Hu, S.: Conditional bert contextual augmentation. In: International Conference on Computational Science. pp. 84–95. Springer (2019)
38. Xie, Z., Wang, S.I., Li, J., Lévy, D., Nie, A., Jurafsky, D., Ng, A.Y.: Data noising as smoothing in neural network language models. In: 5th International Conference on Learning Representations, ICLR 2017 (2017)
39. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)