Text Augmentation Using Dataset Reconstruction for Low-Resource Classification

Adir Rahamim[*]
Technion - Israel Institute of Technology
adir.rahamim@campus.technion.ac.il

Guy Uziel
IBM Research
guy.uziel1@ibm.com

Esther Goldbraich
IBM Research
esthergold@il.ibm.com

Ateret Anaby-Tavor
IBM Research
atereta@il.ibm.com

Abstract

In the deployment of real-world text classification models, label scarcity is a common problem. As the number of classes increases, this problem becomes even more complex. One way to address this problem is by applying text augmentation methods.

One of the more prominent methods involves using the text-generation capabilities of language models. We propose Text AUgmentation by Dataset Reconstruction (TAU-DR), a novel method of data augmentation for text classification. We conduct experiments on several multi-class datasets, showing that our approach improves the current state-of-the-art techniques for data augmentation.

1 Introduction

The deployment of deep learning models in the real-world requires an abundance of labels. However, labeled data is often difficult and expensive to obtain, especially when the models are deployed in highly specialized domains. Therefore, in this paper, we focus on data augmentation for text classification in low-resource environments.

Text classification (Sebastiani, 2002) is fundamental to machine learning and natural language processing. It includes various tasks, such as intent classification (Kumar et al., 2019; Rabinovich et al., 2022), which is a vital component of many automated chatbot platforms (Collinaszy et al., 2017); sentiment analysis (Tang et al., 2015); topic classification (Tong and Koller, 2001; Shnarch et al., 2022); and relation classification (Giridhara et al., 2019). The design and development of such AI applications may begin with a dataset containing only a limited amount of data.

To improve the performance of downstream models in such low-resource settings, a data augmentation mechanism is often implemented (Wong et al., 2016). To achieve this, new data are synthesized from existing training data. It has been demonstrated that the use of such mechanisms can significantly improve the performance of various neural network models. For computer vision and speech recognition, a number of well-established methods are available for synthesizing labeled data and enhancing classification accuracy. Some of the basic methods, which are also class preserving, include transformations such as cropping, padding, flipping, and shifting along time and space dimensions (Cui et al., 2015; Krizhevsky et al., 2017).

However, the application of simple transformation for textual data augmentation is more challenging, since simple transformations often invalidate and distort the text, thereby producing grammatically and semantically incorrect texts that are different from the actual text distribution. Consequently, rule-based data augmentation methods for texts typically involve replacing one word with a synonym, deleting a word, or changing a word (Wei and Zou, 2019; Dai and Adel, 2020).

Recent advances in text generation models (Radford et al., 2018) facilitate an innovative approach for handling scarce data situations. In an effort to reduce the cost of obtaining labeled in-domain data, Wang et al. (2021) use the self-training framework to generate pseudo-labeled training data from unlabeled in-domain data. Xu et al. (2021) have recently demonstrated the difficulty in extracting such domain-specific unlabeled data from general corpora.

A number of existing works (Ding et al., 2020; Anaby-Tavor et al., 2020; Yang et al., 2020) have overcome this difficulty by using the generation capabilities of pre-trained language models.

In this paper, we follow the latter paradigm and propose Text Augmentation by Dataset Reconstruction (TAU-DR), a novel text augmentation algorithm that generates new sentences based on the reconstruction of the original sentences from the

[*] The work was completed during an internship at IBM Research.

Findings of the Association for Computational Linguistics: ACL 2023, pages 7389–7402
July 9-14, 2023 ©2023 Association for Computational Linguistics

hidden representations of a pre-trained classifier.

TAU-DR utilizes frozen auto-regressive language models by soft-prompt tuning, using a relatively small number of trainable parameters compared to the language model and unlike most existing methods that rely on language models, it does not require an additional pertaining phase. During training, we extract the hidden representation from the pre-trained classifier and use a Multi-Layer Perceptron (MLP) to turn the hidden representation into a soft-prompt. The soft-prompt is then fed into the frozen language mode.

Our approach is motivated by the observation that if the pre-trained classifier is trained from a language model (i.e. BERT), then the hidden representation is a contextual embedding of the original sentence. Thus, the soft-prompt will also summarize contextual information from a small neighborhood of the hidden representation, giving the frozen language model additional information for enriching the original dataset.

By using this training approach and manipulating the trained prompts, we are able to generate novel sentences with their corresponding pseudo-labels. Then, as in previous works (Anaby-Tavor et al., 2020; Wang et al., 2022), we apply a filtering mechanism and filter out low-quality sentences.

We conduct experiments on four multi-class datasets: TREC, ATIS, Banking77, and T-Bot (in various low-resource settings) and show that our approach consistently outperforms the current state-of-the-art approaches. We also conduct several experiments measuring the quality of the generated sentences[1].

Our contributions are two-fold, and can be summarized below:

- We propose a novel approach for data augmentation using dataset reconstruction. We demonstrate that our method achieves state-of-the-art performance on several text classification datasets.

- We suggest two novel filtering approaches for better exploitation of the generated sentences - one approach for cases where the evaluation set is available, and another approach for cases where such datasets are absent.

The remainder of the paper is organized as follows: Section 2 introduces the problem framework and relevant studies. In Section 3, we present TAU-DR and our approach. In Section 4, we conduct the experiments. Section 5 concludes the paper and includes a discussion of future work.

2 Problem Setup and Related Work

In this section, we introduce the data augmentation setting in a low-resource text classification. Let $\mathbb{X}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ be a text classification dataset with L classes, where we denote x_i to be the example and y_i to be its corresponding label. We assume that for each class, we have m examples where m is a relatively low number. As in previous works (e.g., Anaby-Tavor et al. 2020; Wang et al. 2022), we assume the existence of a validation set $\mathbb{X}_{\text{val}}$ and a test set $\mathbb{X}_{\text{test}}$.[2]

Our goal is to create an augmented dataset $\mathbb{X}_{\text{gen}}$ by using $\mathbb{X}_{\text{train}}$ so that by training a classifier on the union of the generated and the original dataset $\mathbb{X}_{\text{train}} \cup \mathbb{X}_{\text{gen}}$, we improve the performance of the same classifier trained on $\mathbb{X}_{\text{train}}$. The performance of each classifier is measured on $\mathbb{X}_{\text{test}}$.

The task of text augmentation is relatively challenging, since even small modifications can change the meaning and label of the text. By carefully setting up a rule-based approach, one can deal with this challenge. This was tried by Wei and Zou (2019), who proposed Easy Data Augmentation (EDA), which utilizes simple predefined rules to edit, remove, and substitute portions of the text while maintaining its meaning. Dai and Adel (2020) suggested a rule- based augmentation method named SDANER, tailored for named entity recognition.

A different line of research, which is the prominent approach, uses pre-trained language models. Wu et al. (2019) proposed Conditional BERT (CBERT) for contextual data augmentation. Given a sentence and its label, words in the sentence are masked randomly. The label is then used as a context to predict substitute words while keeping the original sentence in the same class.

Anaby-Tavor et al. (2020) introduced Language Model Based Data Augmentation (LAMBADA), which is also a conditional generation-based data augmentation. LAMBADA fine-tunes an entire language model, $GPT2$, by concatenating all of the sentences together with their corresponding

[1] Our implementation will be released after the anonymity period.

[2] Because this assumption does not hold in some real-world scenarios, in Section 4.5 we abandon that assumption and discuss the no-validation case.

labels, thereby creating additional textual data on which the language model can be fine-tuned. Due to the noisiness of the generation process, a filtering process is used to ensure that only high-quality sentences remain. The filtering process consists of a classifier that was trained on the original dataset by taking those sentences with the top-K softmax scores.

Wang et al. (2022) recently suggested PromDA. This approach first trains an entire pre-trained language model on the task of converting keywords to sentences from a general corpus. Then, using RAKE (Rose et al., 2010), keywords are extracted from the original dataset. By concatenating these keywords to a learned prefix, the language model from the previous step is used to reconstruct the original sentence. Then, the same filtering process as in LAMBADA is used, with the exception that *all* sentences for which the original classifier agrees with the pseudo-label are taken.

2.1 Soft-Prompts

TAU-DR, as will be discussed in the next section, exploits the language-generation capabilities of language models by using soft-prompts, one of the dominant approaches for parameter-efficient tuning. Prompt-based learning was introduced by Brown et al. (2020). Their study demonstrated that a large language model can be adapted for downstream tasks by carefully constructing prompts (i.e., textual instructions). A method proposed by Gao et al. (2020) for simplifying the construction process involves expanding prompts by using pretrained language models. Each downstream task requires manual construction of discrete prompts. The construction of discrete prompts is still an independent process that is difficult to optimize together with downstream tasks.

A study by Lester et al. (2021); Li and Liang (2021) suggests using soft-prompts. Soft-prompts do not represent actual words, as opposed to hard prompts, and can be incorporated into frozen pre-trained language models. As demonstrated by Li and Liang (2021), pre-trained language models (PLMs) with soft-prompts provide better performance in low-resource settings, and enable end-to-end optimization of downstream tasks.

,

3 Text Augmentation by Dataset Reconstruction (TAU-DR)

In this section, we introduce Text AUgmentation by Dataset Reconstruction (TAU-DR), our novel text augmentation algorithm.

Algorithm 1 Text Augmentation by Dataset Reconstruction (TAU-DR)

```
Require: Training dataset $\mathbb{X}_{\text{train}}$, pre-trained classifier $\mathbb{C}_{base}$, pre-trained language model $\mathcal{LM}$
          %% training phase
 1: while training steps not done do
 2:     for $(x, y)$ in $\mathbb{X}_{\text{train}}$ do
 3:         Extract $h$ from $\mathbb{C}_{base}$
 4:         $P \leftarrow MLP(h)$ % transform the hidden representation into soft-prompt.
 5:         $\hat{x} \leftarrow \mathcal{LM}(P)$ % predict a sentence using the soft-prompt
 6:         $\theta_{MLP} \leftarrow \theta_{MLP} - \nabla_{\theta_{MLP}} \mathcal{L}_{lm}(x, \hat{x})$
 7:     end for
 8: end while
          %% generation phase
 9: $\mathbb{X}_{\text{intra}} \leftarrow GEN_{intra}(\mathcal{LM}, MLP, \mathbb{X}_{\text{train}})$
10: $\mathbb{X}_{\text{inter}} \leftarrow GEN_{inter}(\mathcal{LM}, MLP, \mathbb{X}_{\text{train}})$
11: $\mathbb{X}_{\text{gen}} \leftarrow \mathbb{X}_{\text{intra}} \cup \mathbb{X}_{\text{inter}}$
          %% filtration phase
12: $\mathbb{X}_{\text{gen}} \leftarrow Filtration(\mathbb{X}_{\text{train}}, \mathbb{X}_{\text{val}}, \mathbb{X}_{\text{gen}})$
```

TAU-DR consists of three stages: training, generation, and filtration, as described below.

3.1 Training

We now describe the training phase in TAU-DR as shown in Algorithm 1. Given an example x from the original dataset, we extract its hidden representation, h, from the pre-trained classifier, which we denote by C_{base} (line 3). For instance, if C_{base} is a BERT classifier, it can be the $[CLS]$ token representation in the last layer. The next step in line 4 is to apply a multi-layer perceptron (MLP) with parameters θ_{MLP}, and turn the hidden representation, h, into a prompt of length n denoted as P. P is then fed into the frozen language model $\mathcal{LM}$ (line 5). The training objective of the language model is to reconstruct the original sentence using only the hidden representation. The training step is illustrated in Figure 1.

3.2 Generation

To generate new sentences that will challenge the classifier and ultimately improve its accuracy, we

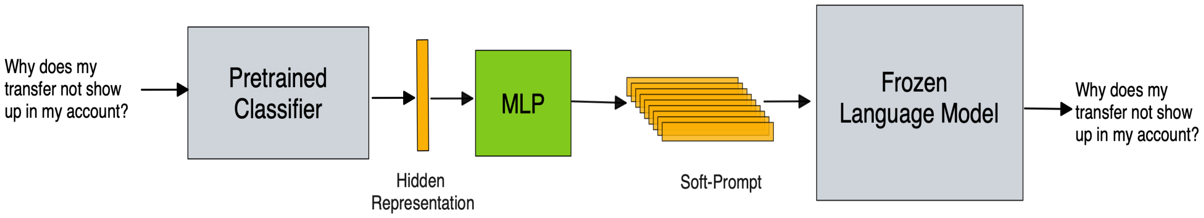
Why does my transfer not show up in my account?
Pretrained Classifier
Hidden Representation
MLP
Soft-Prompt
Frozen Language Model
Why does my transfer not show up in my account?

Figure 1: TAU-DR: We take a sentence from the dataset and pass it through the pre-trained classifier. Then, the last hidden representation (i.e., [cls] token) is used as an input for the multi-layered preceptron (MLP), whose parameters are the only trained parameters. The MLP outputs a soft prompt and the generator reconstructs the original sentence.

perturb the learned soft prompts. We suggest two novel strategies to provide new soft prompts for the frozen language model[3].

Intra-class generation The motivation behind the following approach is that by combining soft prompts from the same class, we will be able to lexically and semantically enrich the class itself. The method can be described as follows: We select two sentences, x_1, x_2 from the same class, and then extract their corresponding hidden representation, h_1, h_2, using the pre-trained classifier C_{base}. Using the trained MLP, we transform them into their corresponding soft prompts, P_1 and P_2. Then, by averaging the two prompts, we achieve a new aggregated soft prompt P_{agg}. The latter is passed into the language model. The pseudo-label for the generated sentences is set as class x_1. This is illustrated in Figure 2.

Inter-class generation With inter-class generation, we help the classifier to better distinguish between the different classes. This is done by generating sentences using soft-prompts, which are created by combining soft-prompts from two different classes. First, we randomly sample two sentences from two different classes, x_1, x_2, and then, as detailed above, extract their soft-prompts denoted as P_1 and P_2, respectively. We then aggregate the two prompts by taking their weighted mean, $P_{agg} = wP_1 + (1-w)P_2$, where $0 < w < 1$ is sampled uniformly. In this case, we set the pseudo-label as the label of the closest prompt as illustrated in Figure 3.

3.3 Dynamic Consistency Filtering

By generating new sentences for our classifier, we risk the creation of low-quality data. This can happen if we set an incorrect pseudo-label or if the language model generates out-of-domain examples. Therefore, it is common to apply a consistency filtering mechanism (Anaby-Tavor et al., 2020; Wang et al., 2022).

The consistency filtering suggested by Anaby-Tavor et al. (2020) used the pre-trained classifier and considered the top-K sentences (ordered by their softmax scores). Wang et al. (2022) also used the trained classifier. However, instead of using the top-K approach, they kept all the generated sentences for which the classifier agrees with the pseudo-label.

Clearly, the chosen filtration method has a large effect on the final classifier, as it controls the data quality of the final trained classifier. The top-K approach might be too conservative, keeping a large safety margin, which results in filtering out most of the generated instances. On the other hand, keeping all the instances on which the classifier agrees with the pseudo-label might include many noisy-label sentences, resulting in a degraded classifier.

We now present *Dynamic Consistency Filtering* - our filtering approach for a case where an evaluation set exists. In Section 4.5, we discuss the no- evaluation case. Our method relies upon the evaluation dataset to approximate the optimal portion of the generated instances to include in the augmented dataset. We do so by training k classifiers, one of which trained on a different quantile of the generated instances, ordered by their softmax scores (received from the pre-trained classifier C_{base}). After training the k instances, we choose the best preforming classifier using the evaluation dataset.

It is important to note that there is a possibility of applying the filtering mechanism in a recursive manner, for example, training a classifier on the filtered data and running that classifier on the original

[3] The closest work to this approach is the work of Asai et al. (2022), suggesting the aggregation of prompts for multitask generalization.

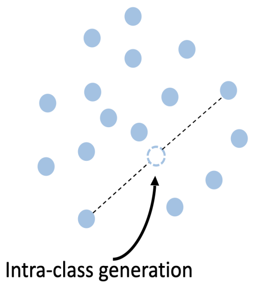

Figure 2: Intra-class generation: We sample two instances from the same class and average their prompts. The averaged prompt is then used as a prompt for the generator. The (pseudo)-label is decided according to the class of the instances.

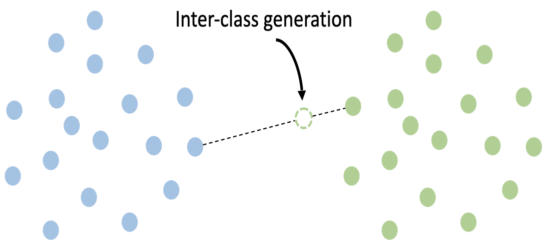

Figure 3: Inter-class generation: We sample two instances from two different classes. We then calculate the weighted average of their prompts (the weights are sampled randomly). The (pseudo)-label is decided according to the closest sample.

generated dataset, with the hope of improving the filtering of the instances. This way, one can further improve the performance of the final classifier, as discussed by Anaby-Tavor et al. (2020).

3.4 Training and Generating in Low-Resource Setting

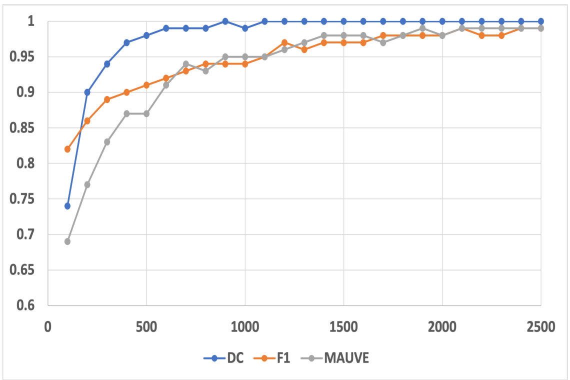

Figure 4: F1, DC and MAUVE of the generated text distribution compared to the train distribution as a function of the training steps.

In a preliminary study we conducted, we investigated how over-training affects the quality of the generated text in terms of diversity and distance from the original train distribution. To show the effect of over-training, we apply LAMBADA (Anaby-Tavor et al., 2020) on several internal low-resource multi-class datasets. We trained LAMBADA for 2500 steps and generated augmented sentences every 100 steps. We then evaluated the quality of the generated sentences as a function of the trained steps by using distributional measures: Precision and Recall (Sajjadi et al., 2018) summarized as F1, DC (Naeem et al., 2020) and MAUVE (Pillutla et al., 2021) [4].

We can observe, on Figure 4, that DC, Precision and Recall and MAUVE converges to 1. This suggest that without any control measures in place, the distribution of the generated text quickly converges into the training distribution. This is not a desired property since our goal is to generate texts which will expand the support of the training distribution. It is interesting to note that the nature of the results remains the same, even when soft-prompt tuning is applied.

Therefore, to address the above, we deploy two heuristics. The first heuristic is to increase the number of training samples. We do so by using the EDA rule-based simple augmentation method discussed earlier (Wei and Zou, 2019). Please note that in this enrichment we do not consider the pseudo-labels, since our goal is to provide more reference points for the MLP training. Moreover, we checkpoint the MLP several times during training, and generate sentences from the different checkpoints.

4 Experiments

4.1 Setup

We conduct experiments on four multi-class classification datasets (described in the next subsection). Each benchmark dataset is split into 80% train ,10% evaluation and 10% test. We then take the train dataset and sample K examples for each class where classes without K examples are removed, resulting in a shot-K dataset. In our experiments, we choose $K \in (5, 10)$. As a base classifier, we choose the BERT-base model[5], as in the study of Anaby-Tavor et al. (2020); Wang et al. (2021).

[4] The measures are introduced on Section 4.4.

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

	Shot-5				Shot-10			
Method	ATIS	TREC	Banking77	T-Bot	ATIS	TREC	Banking77	T-Bot
C_{base}	0.739	0.495	0.689	0.681	0.772	0.713	0.798	0.741
EDA	0.735	0.524	0.7	0.684	0.806	0.72	0.793	0.749
C-BERT	0.75	0.517	0.682	671	0.877	0.727	0.805	0.747
LAMBADA	0.88	0.566	0.709	0.703	0.871	0.745	0.787	0.74
PromDA	0.867	0.583	**0.739**	0.692	0.897	0.742	0.791	0.752
TAU-DR	**0.906**	**0.641**	0.733	**0.71**	**0.933**	**0.773**	**0.839**	**0.788**

Table 1: The average accuracy results of the different benchmarks to the multi-class classification tasks. The best improvement in each configuration over the performance of the base model is marked in bold. The results of TAU-DR are significant compared to C_{base} (paired student's t-test, $p < 0.05$).

The same set of hyperparameters is used for the training of C_{base}, for training without the original data, and for training with the generated data. The performance of C_{base} is evaluated during training using $\mathbb{X}_{val}$.

We compare TAU-DR to the methods discussed in Section 4.1: The rule-based data augmentation methods EDA (Wei and Zou, 2019); CBERT (Wu et al., 2019), LAMBADA (Anaby-Tavor et al., 2020), and PromDA (Wang et al., 2022) which is implemented with a T5-large model (700M parameters). All hyperparameters used for these methods are those recommended by the authors. We repeat the experiments five times and report the averaged accuracy for each shot-k dataset.

For TAU-DR we used the T5-large model for all of our experiments. This model was fine tuned an additional $100k$ steps on the C4 dataset using the regular LM loss, to achieve better adaptivity to soft prompt tuning (Lester et al., 2021)[6]. We choose MLP with 2 hidden layers and a ReLU activation. The prompt-length is set as 10 in all of our experiments. TAU-DR was trained for 100 epochs. We checkpointed the model every 20 epochs, resulting in 5 checkpoints. The pre-trained classifier C_{base} used in our method is the same classifier discussed above. For the dynamic filtering, we use 10 classifiers with the same configuration as the pre-trained classifier, where each classifier is trained on a different portion of the generated dataset ordered by the softmax score of C_{base}. The experimental results are shown in Table 1 for shot-5 and shot-10 for the different multi-class benchmarks.

4.2 Datasets

All datasets used are classification datasets, with different numbers of classes and across several domains, three of which are available in the public domain.

Name	# Classes	Domain
ATIS	17	Flight reservation
TREC	50	Open-domain questions
Banking77	77	Banking
T-Bot	87	Telco customer support

Table 2: Properties of the used multi-class datasets

Airline Travel Information Systems (ATIS, Hemphill et al. 1990): The ATIS dataset provides a large set of queries about flight information along with the intent, the subject of the various questions.

Text Retrieval Conference (TREC, Hovy et al. 2001): TREC is a question classification dataset that consists of a variety of questions from different areas and their intent.

Banking77 (Casanueva et al., 2020): The Banking77 dataset offers questions from single-domain banking, annotated with their labels.

Teleco-Bot (T-Bot): An internal intent classification dataset, includes data used for the training of chatbots used by telco companies for customer support.

The datasets used are summarized in Table 2

4.3 Main results

First, we can observe that the addition of the generated data from TAU-DR to the classification models significantly improves the performance of C_{base} and outperforms the existing method. Overall, the EDA rule-based approach does not lead to a significant improvement over C_{base} on the more challenging datasets Banking77 and T-Bot on both shot-5 and shot-10. On the other hand, the

[6] https://huggingface.co/google/t5-large-lm-adapt

language-model-based approaches, i.e., C-BERT, LAMBADA, PromDA and TAU-DR outperform the rule-based approach. PromDA can provide better results than LAMBADA on the ATIS and TREC datasets. However, with the exception of Banking77 (shot-5) it fails when considering domain-specific datasets with a larger number of classes, such as Banking77 and T-Bot.

On T-Bot and Banking77 in the shot-10 setting none of the methods expect TAU-DR where able to give a statistically significant improvement over C_{base}.

The accuracy improvements of TAU-DR over C_{base} on the ATIS dataset are approximately 20% for both shot-5 and shot-10. For TREC the improvement rate is 29% for the shot-5 setting and 9% for the shot-10 setting. For Banking77, the average improvement rate is 4.5% and for the challenging T-Bot dataset the average improvement rate is 5%.

4.4 Estimating the Generation Quality

We now turn to estimate the quality of text generated by the different methods. We use the following measures:

- Recall and Precision (Sajjadi et al., 2018): Given two distributions P, Q, this measure compares their "precision", or how much of Q can be generated by a "part" of P, while "recall" measures how much of P can be generated by a "part" of Q. Recall and Precision are summarized as F1.

- Complexity (Kour et al., 2021): Quantifies how difficult observations are, given their true class label and how they will challenge the classifier. The measure can be used to automatically determine a baseline performance threshold..

- MAUVE (Pillutla et al., 2021): This metric measures the gap between two text distributions by calculating the area under the information divergence curve.

A recent study (Kour et al., 2022) compared several statistical and distributional measures. The different measures were compared over several desired criteria. In their experiments, MAUVE turned out to be the most robust performance measure for text generation quality.

In this set of experiments, we took the generated text and compared it to *the test set*, which represents the actual text distribution. A desired property of the augmented texts is that their distribution will expand the intersection between the support of the train distribution with the test distribution. Thus, we can compare the generation quality of the different methods by looking on how close they are to the test distribution.

We report the average results on Table 3. The implementation details for this experiment are detailed on Appendix B. The measures of the text generated by TAU-DR are superior to 2 out of 3 in all configurations. Showing that we can generate text that is close to the actual distribution of the data. In addition, by looking at the Atis dataset, we observe that we were able to produce more challenging and complex sentences for the classifier.

It has not been explored if or how these measures relate to the classifier's performance. Nevertheless, these measures can provide some insight into how well a model can reproduce the test distribution.

4.5 Dynamic Consistency Filtering with no Evaluation

Our suggested dynamic filtering method is shown to be effective in filtering out low-quality generated data. However, the existence of such datasets is not obvious in real-world scenarios.

In this subsection, we suggest an approach for filtering the generated data without relying on the existence of an evaluation dataset. The method can be described as follows: As in the Dynamic Consistency Filtering approach, for each class we order the generated examples according to their softmax scores obtained from the pre-trained classifier C_{base}. We then filter out all instances on which the classifier disagrees with the pseudo-label. Then we train k classifiers on a different quantile of the ordered data (i.e., for $k = 5$, we train the i-th classifier $i = 1, ..., 5$, on the $i/5$ quantile). We then use the obtained classifiers to filter out the generated instances based on the majority vote of the classifiers, we denote this approach as TAU-DR$_{maj}$. As shown in Table 4, with the exception ATIS (shot-5) and Banking77 (shot-5), TAU-DR$_{maj}$ also outperforms the benchmark methods and on average only slightly degrades the performance of TAU-DR.

	Shot-5			Shot-10		
Method	F1↑	MAUVE↑	Complexity↑	F1↑	MAUVE↑	Complexity↑
	ATIS			ATIS		
LAMBADA	0.66	**0.78**	10.01	**0.76**	0.8	5.43
PromDA	0.7	0.71	8.44	0.74	0.76	4.67
TAU-DR	**0.75**	0.7	**13.67**	0.73	**0.82**	**5.91**
	Banking77			Banking77		
LAMBADA	0.79	0.72	**2.67**	0.86	0.8	2.56
PromDA	0.83	0.75	3.66	0.85	0.75	**3.65**
TAU-DR	**0.86**	**0.78**	2.61	**0.88**	**0.87**	2.31

Table 3: The average of the generation quality measures two of the multi-class classification tasks, ATIS and Banking77. The best performing approach in each configuration is marked in bold.

	Shot-5				Shot-10			
Method	ATIS	TREC	Banking77	T-Bot	ATIS	TREC	Banking77	T-Bot
BASE	0.739	0.495	0.689	0.681	0.792	0.713	0.798	0.741
TAU-DR	0.906	0.641	0.733	0.71	0.933	0.773	0.839	0.788
TAU-DR$_{maj}$	0.847	0.615	0.738	0.726	0.911	0.761	0.833	0.767

Table 4: The average accuracy obtained by the base classifier C_{base} and TAU-DR with the dynamic filtering approach and with TAU-DR equipped with the weighted majority filtration approach not relying on the existence of evaluation dataset.

5 Conclusion and Future Work

In this paper, we present TAU-DR, a novel text-augmentation method for low-resource classification using dataset reconstruction. We test our method on four multi-class classification datasets in various few-shot scenarios and show that our approach outperforms the state-of-the-art approaches. In the future, we plan to explore the learned prompt space and check how it can be used for generating helpful sentences. In our preliminary experiment, we found that the averages of the prompts were concentrated in a narrow cone. This concentration hinders the exploitation of the geometry in the learned prompt space. The above observation is aligned with other findings regarding the anisotropy of the word embedding space in pre-trained language models (Li et al., 2020; Ethayarajh, 2019). Finally, we wish to explore if and how additional information (e.g, in-domain textual-data) might improve the performance of text augmentation methods on highly specialized domains.

Limitations

To address the low-resource data in the training of TAU-DR, we apply two heuristics, dataset enrichment and generation from different checkpoints. Despite being effective, they require additional computational time that might be challenging in applications with low-computational resources. A possible approach to reduce the computational time might be to average the checkpoints. We believe that this might lead to competitive results, with a significant reduction in computational time, since checkpoint averaging proved to be an effective approach in low-resource settings. Another limitation is when the original dataset is in a highly-specialized domain that might contain domain-specific phrases that were most likely not included in the pre-training data of the language model. The results obtained by existing data augmentation approaches will most likely exhibit only marginal improvement.

Ethics Statement

Text generation by nature entails a number of ethical considerations when considering possible applications. The main failure is when the model generates text with undesirable properties (bias etc.) for training the classifier but these properties are not present in the original training data.

Because our model converges and learns to generate data close to the underlying source material, the above considerations, in our approach, are negligible. As a result, the generated text may be harmful if users of such models are unaware that such issues appear on their training data or if they fail to consider them, e.g., by selecting and evaluating data more carefully.

References

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

Akari Asai, Mohammadreza Salehi, Matthew E Peters, and Hannaneh Hajishirzi. 2022. Attentional mixtures of soft prompt tuning for parameter-efficient multi-task knowledge sharing. *arXiv preprint arXiv:2205.11961*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Juraj Collinaszy, Marel Bundzel, and Iveta Zolotova. 2017. Implementation of intelligent software using ibm watson and bluemix. *Acta Electrotechnica et Informatica*, 17(1):58–63.

Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury. 2015. Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(9):1469–1477.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. *arXiv preprint arXiv:2010.11683*.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. Daga: Data augmentation with a generation approach for low-resource tagging tasks. *arXiv preprint arXiv:2011.01549*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Praveen Kumar Badimala Giridhara, Chinmaya Mishra, Reddy Kumar Modam Venkataramana, Syed Saqib Bukhari, and Andreas Dengel. 2019. A study of various text augmentation techniques for relation classification in free text. *ICPRAM*, 3:5.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

George Kour, Samuel Ackerman, Orna Raz, Eitan Farchi, Boaz Carmeli, and Ateret Anaby-Tavor. 2022. Measuring the measuring tools: An automatic evaluation of semantic metrics for text corpora. *arXiv preprint arXiv:2211.16259*.

George Kour, Marcel Zalmanovici, Orna Raz, Samuel Ackerman, and Ateret Anaby-Tavor. 2021. Classifier data quality: A geometric complexity based method for automated baseline and insights generation. *arXiv preprint arXiv:2112.11832*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. 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 3609–3619.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. 2020. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pages 7176–7185. PMLR.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Ella Rabinovich, Matan Vetzler, David Boaz, Vineet Kumar, Gaurav Pandey, and Ateret Anaby-Tavor. 2022. Gaining insights into unrecognized user utterances in task-oriented dialog systems. *arXiv preprint arXiv:2204.05158*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, pages 1–20.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. 2018. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Eyal Shnarch, Alon Halfon, Ariel Gera, Marina Danilevsky, Yannis Katsis, Leshem Choshen, Martin Santillan Cooper, Dina Epelboim, Zheng Zhang, Dakuo Wang, et al. 2022. Label sleuth: From unlabeled text to a classifier in a few hours. *arXiv preprint arXiv:2208.01483*.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. Promda: Prompt-based data augmentation for low-resource nlu tasks. *arXiv preprint arXiv:2202.12499*.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Sebastien C Wong, Adam Gatt, Victor Stamatescu, and Mark D McDonnell. 2016. Understanding data augmentation for classification: when to warp? In *2016 international conference on digital image computing: techniques and applications (DICTA)*, pages 1–6. IEEE.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International conference on computational science*, pages 84–95. Springer.

Xinnuo Xu, Guoyin Wang, Young-Bum Kim, and Sungjin Lee. 2021. Augnlg: Few-shot natural language generation using self-trained data augmentation. *arXiv preprint arXiv:2106.05589*.

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 preprint arXiv:2004.11546*.

A Ablation

In our ablation studies, we evaluated the independent effect of 5 different components on our method: enrichment, intra-class, inter-class, checkpointing and dynamic filtering. In this ablation study we want to emphasize the contribution of each module for the success of our method. Results are summarized in Table 5. During the MLP training we used dataset enrichment in order to add more reference points. As we can observe from the results this enrichment is an important aspect of our method as our method without dataset enrichment results in an average degradation of 4.5 accuracy points. In addition, we evaluated the effect of each generation method we proposed – intra-class and inter-class. The intra-class generation is meant to enrich the number of examples in a given class, whereas inter-class is meant to highlight the difference between different classes. We can see that both generation methods are vital components of our method, with degradation of 2 and 3.25 accuracy points when not using intra-class or inter-class, respectively.

Moreover, we determined the efficacy of the checkpointing paradigm. We utilized checkpointing to overcome the over training affects as discussed on Section 3. Based on the results, we can see that the checkpointing paradigm plays an important role in the method's success. Generating sentences using only the last checkpoint results in degradation of 3.75 accuracy points. The last ablation conducted is to evaluate the performance of the dynamic filtering method. As discussed earlier in all the generation methods this plays a vital component in keeping high-quality instances. On the ablation experiment we kept all the sentences on which C_{base} agrees with the pseudo-label. Not surprisingly this approach caused a major decrease in the accuracy with an average of 6.5 accuracy points.

B Additional implementation details

The optimizer used for training the MLP is AdamW, we tested the following learning rate $\{1e-3, 1e-2, 1e-4\}$ and a $1e-2$ weight decay. We experimented with the following batch sizes $\{16, 32, 64\}$. he size of the hidden layer is set as $dim(h) * n/2$, where n is the prefix-length. The MLP architecture was not optimized during our experiments. We experiment also with prefix-lengths of $5, 10, 15, 20$. These different prefix lengths have a negligible impact, since we used a medium-sized model. This aligns with the findings of Lester et al. (2021) . We used an internal multi-class dataset which was not reported in the main paper to search for the best training configuration. The classifier was trained for 5000 steps with 8 batch size with AdamW optimizer and $1e-5$ learning rate We run all experiments on a single NVIDIA A100 GPU. For the generation phase, we used the nucleus sampling (Holtzman et al., 2019) with $k = 100, p = 0.95$ both for the intra- and inter-generation approaches.

To calculate Precision and Recall, MAUVE and Complexity we sampled 1000 instances and compared against 1000 sentences in the generated set. We repeated this process 10 times for every one of the 5 splits for each dataset.

	Shot-5			
Method	ATIS	TREC	BANKING77	WVA
TAU-DR	0.906	0.641	0.733	0.71
TAU-DR w/o enrichment	0.858	0.596	0.713	0.679
TAU-DR w/o intra-gen.	0.894	0.617	0.728	0.672
TAU-DR w/o inter-gen.	0.837	0.631	0.702	0.695
TAU-DR w/o checkpointing	0.875	0.602	0.717	0.694
TAU-DR w/o dynamic filtering	0.761	0.57	0.697	0.708

Table 5: The average accuracy results of the different components of TAU-DR on the multi-class classification tasks.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section number 5

☑ A2. Did you discuss any potential risks of your work?
Section number 5

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 4

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix B and Section 4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4 and Appendix B

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.