Guiding Generative Language Models for Data Augmentation in Few-Shot Text Classification

Aleksandra Edwards† **Asahi Ushio**† **Jose Camacho-Collados**†
Hélène de Ribaupierre† **Alun Preece**‡
†School of Computer Science and Informatics, Cardiff University, United Kingdom
‡Crime and Security Research Institute, Cardiff University, United Kingdom
{edwardsai,ushioa,camachocolladosj,deribaupierreh,preecead}@cardiff.ac.uk

Abstract

Data augmentation techniques are widely used for enhancing the performance of machine learning models by tackling class imbalance issues and data sparsity. State-of-the-art generative language models have been shown to provide significant gains across different NLP tasks. However, their applicability to data augmentation for text classification tasks in few-shot settings have not been fully explored, especially for specialised domains. In this paper, we leverage GPT-2 (Radford et al., 2019) for generating artificial training instances in order to improve classification performance. Our aim is to analyse the impact the selection process of seed training examples have over the quality of GPT-generated samples and consequently the classifier performance. We perform experiments with several seed selection strategies that, among others, exploit class hierarchical structures and domain expert selection. Our results show that fine-tuning GPT-2 in a handful of label instances leads to consistent classification improvements and outperform competitive baselines. Finally, we show that guiding this process through domain expert selection can lead to further improvements, which opens up interesting research avenues for combining generative models and active learning.

1 Introduction

Data sparsity and class imbalance are common problems in text classification tasks (Türker et al., 2019; Zhang and Wu, 2015; Shams, 2014; Kumar et al., 2020), especially when the text to be labelled is from a highly-specialised domain where only scarce domain experts can perform the labelling task (Türker et al., 2019; Ali, 2019). Data Augmentation (DA) is a widely used method for tackling such issues (Anaby-Tavor et al., 2020; Kumar et al., 2020; Papanikolaou and Pierleoni, 2019). However, the well-established DA methods in domains such as computer vision and speech recognition (Anaby-Tavor et al., 2020; Giridhara et al., 2019; Krizhevsky et al., 2017; Cui et al., 2015; Ko et al., 2015; Szegedy et al., 2015), relying on simple transformations of existing samples, cannot be easily transferred to textual data as they can lead to syntactic and semantic distortions to text (Giridhara et al., 2019; Anaby-Tavor et al., 2020).

Recent advances in text generation models, such as GPT and subsequent releases (Radford et al., 2018), have led to the development of new DA approaches which generate additional training data from original samples, rather than perform only local changes to the text. Related studies use text generation models for improving relation extraction (Papanikolaou and Pierleoni, 2019; Kumar et al., 2020), tackle class imbalance problems for extreme multi-label classification tasks (Zhang et al., 2020), and augment domain-specific datasets in order to improve performance in various domain-specific classification tasks (Amin-Nejad et al., 2020). Specifically, Kumar et al. (2020) and Anaby-Tavor et al. (2020) explore different fine-tuning approaches for pre-trained models for data augmentation in order to preserve class-label information. Results showed the potential of generative models such as GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2019) to augment small collections of labelled data. However, these studies present findings for a limited range of classification settings and datasets. Further, an important problem with text generation techniques is the possibility of generating noise which decreases the performance of classification models rather than improving it (Yang et al., 2020).

Therefore, our aim is to improve the quality of generated artificial instances and thus improve classifiers by developing seed selection strategies to guide the generation process. Specifically, we propose three simple DA methods in order to improve few-shot text classification performance using GPT-2 — 1) a method that leverages the expert-generated classification hierarchy of a dataset in order to improve the classification of the top hierarchy classes; 2) a method that selects the seeds with the maximum occurrence of nouns; and 3) a method that involves a domain expert choosing class representative samples. We chose these seed selection strategies because they exploit characteristics associated with specialised domains such as high number of terms, annotation performed by experts, and hierarchical class structure (common for social science and medical domains which require thematic analysis). Moreover, we analyse how different approaches of fine-tuning GPT-2 affect the quality of generated data and consequently the classification performance. Specifically, we identify which fine-tuning method leads to generating higher quality data – fine-tuning on smaller but labelled instances or fine-tuning on a larger unlabelled collection.

Our results show that fine-tuning a GPT-2 model per label improves performance of classifiers consistently, compared to the same model fine-tuned on the entire dataset. Another important conclusion that our study shows and that should be explored further is how guiding this process through domain expert selection in the original dataset can lead to further improvements.

2 Data Augmentation: Related Work

The task of data augmentation consists of generating synthetic additional training samples from existing labelled data (Anaby-Tavor et al., 2020). In the following, we describe standard text augmentation methods which we use as baselines. We also explain recent DA methods based on text generation models.

Word replacement-based (WR). Simple but commonly used DA techniques are based on word-replacement strategies using knowledge bases (Wei and Zou, 2019) such as WordNet (Miller, 1998). Such methods often struggle to preserve the class label and lead to grammatical distortions of the data (Kumar et al., 2020; Giridhara et al., 2019; Anaby-Tavor et al., 2020). Recent DA approaches address the above issues by using language models to provide more contextual knowledge such as CBERT (Wu et al., 2019) in the word replacement process. However, methods that make only local changes to given instances produce sentences with a structure similar to the original ones and thus lead to low variability of training instances in the corpus (Anaby-Tavor et al., 2020).

Sentence replacement-based (SR). Common sentence replacement-based methods are based on back-translation strategies where a given sentence is translated to a language and then back to the original language in order to change the syntax but not the meaning of the sentence (Sennrich et al., 2016; Fadaee et al., 2017). For instance, an original input English sentence is translated to German and then back to English in order to create additional training data.

Text Generation (TG). Recent language models such as GPT-2 (Radford et al., 2019) can address the issues associated with the previous strategies by generating completely new instances from given seed samples. GPT-2 was trained with a causal language modeling (CLM) objective which makes it suitable for predicting the next token in a sequence. This model has been used successfully in text generation tasks such as summarising (Xiao et al., 2020; Kieuvongngam et al., 2020; Alambo et al., 2020) and question answering (Liu and Huang, 2019; Baheti et al., 2020; Klein and Nabi, 2019). Previous research on using text generation techniques for DA for text classification focused on the creation of label-preservation techniques for the generated synthetic data samples and comparing different TG techniques (Anaby-Tavor et al., 2020; Wang and Lillis, 2019; Zhang et al., 2020; Kumar et al., 2020). However, these works are limited in scale and solutions for improving quality of generated data. Further, there is no study on the importance of fine-tuning text generation models to ensure label preservation.

The most similar study to ours is probably that of Yang et al. (2020), in the context of commonsense reasoning. They proposed an approach based on the use of influence functions and heuristics for selecting the most diverse and informative artificial samples from an already-generated artificial dataset. Instead, we focus on the previous step of selecting the most informative samples (or *seeds*) from the original data. We show that a careful selection of class representative samples from the original data in the first place can already lead to important improvements and has an important efficiency advantage, as it prevents an unnecessary waste of resources and time of generating unused generated documents, especially considering how resource expensive generative language models are (Strubell et al., 2019; Schwartz et al., 2019).

3 Methodology

In this paper, we focus on few-shot text classification and, in particular, the DA component. Our

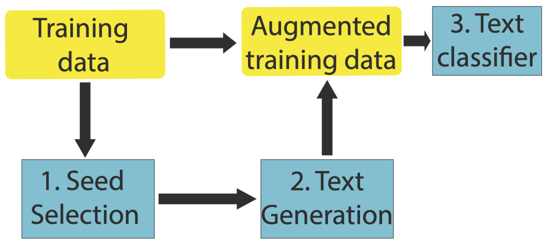

Figure 1: Overview of the methodology

DA methodology consists of three main steps (see Figure 1). In the first step, *Seed Selection*, we select samples (i.e., seeds) from the original labeled data based on four different strategies (Section 3.1). In the second step, *Text Generation*, we generate additional artificial training data using a generative language model using three strategies to fine-tune it on different input texts (Section 3.2). Finally, the augmented training data is used in combination with the original data to train a *Text Classifier* (Section 3.3).

3.1 Seed Selection Strategies

We implement four seed selection strategies, which we describe below.

Random Seed Selection. For this strategy we simply select a fixed number of instances in a random manner. We use random selection to evaluate whether the rest of the seed selection strategies lead to improvements in classification.

Maximum Nouns-guided Seed Selection. Many specialised domains are rich of domain-specific terminology and thus we believe that noun-rich instances might be more indicative for the classes compared to the other training samples. Therefore, we use this strategy to select the seeds with the maximum occurrence of nouns. We identify single word nouns and compound nouns within data using NLTK (Bird and Loper, 2004).

Subclass-guided Seed Selection. In this strategy, we leverage the human-generated classification hierarchy of a dataset in order to improve the classification of the top classes. Specifically, we select a roughly balanced number of seeds from each subclass belonging to a given label. In this way, we diversify the vocabulary for each overall class by ensuring the equal participation of representative samples from even the most underrepresented subclasses.

Expert-guided Seed Selection. The highly specialised nature of some domains where the manual annotation of documents is performed by experts show that identifying class representative samples might require more implicit knowledge that is hard to be captured by statistical approaches. Therefore, in the *Expert-guided Seed Selection* strategy, we conducted a study asking experts to select the class representative samples from the original training data. The chosen seeds are then used to generate additional training data. We explain how the case study was conducted in Section 4.5.

3.2 Text Generation

We generate artificial data using the generative pre-trained model, GPT-2 (Radford et al., 2019). We use GPT-2 model as it gives a state-of-the-art performance for many text generation tasks and also have been designed with the objective to fit scenarios with few-shot and even zero-shot settings. We use two methods for fine-tuning the GPT-2 model — we fine-tune the model on the entire dataset and we also fine-tune a specific GPT-2 model for each given class to ensure label-preservation for the generated sequences. We also perform experiments using a pre-trained GPT-2 model. We compare three three models in order to assess the need of fine-tuning and the use of additional methods for label-preservation when using TG-based DA for classification tasks. These models are then leveraged to generate new documents given a labeled instance. To ensure robustness, the text generation step is performed for three iterations and the results are averaged.

3.3 Text Classification

In this final step, we use the augmented training data to train a fastText classifier (Joulin et al., 2017) coupled with domain-trained fastText word embeddings. The reason to use a simple model such as fastText is its efficiency and that transformer-based models tend to not perform well with limited data in document classification and in general tasks that do not require a fine granularity (Joshi et al., 2020). Indeed, fastText has been shown to perform equally or better with limited labeled data in document classification, compared to more sophisticated models such as BERT (Edwards et al., 2020). Moreover, the main aim of our paper is not to achieve state-of-the-art text classification performance, but to compare the quality of the generated outputs in the context of few-shot text classification.

4 Experimental Setting

In the following we describe our few-shot text classification experimental setting.[1]

4.1 Datasets

For our experiments we selected a suite of datasets from different domains which all have a hierarchical classification structure. These are: *20 Newsgroups* (Lang, 1995), *Toxic comments* (Hosseini et al., 2017), and *Safeguarding reports* (Edwards et al., 2021). The 20 Newsgroups collection is a popular data set for experiments in machine learning. The data is organized into 20 different newsgroups, each corresponding to a different news topic such as computer systems, religion, politics (Lang, 1995). The collection of the Toxic comments dataset is obtained from Wikipedia and it is the result from the collaboration between Google and Jigsaw for creating a machine learning-based system for automatically detecting online insults, harassment, and abusive speech (Hosseini et al., 2017). The purpose of the safeguarding reports is to identify and describe related events that precede a serious safeguarding incident and to reflect on agencies' roles. The classification is performed over themes (e.g. mental health issues, indicative circumstances) at the sentence level. As a special trait of this dataset, the reports contain domain-specific terminology which makes them hard to analyse with existing text analysis tools (Edwards et al., 2019).

We perform prediction for the top classes of each dataset. However, as mentioned in Section 3, we use the sub-classes to select seed instances. For simplicity and setting unification, we convert the multi-label classification tasks of the *Toxic comments* and *Safeguarding* datasets to multi-class problems, removing the few instances that were labeled with more than one class in the original datasets. The main features and statistics of each dataset are summarized in Table 1.[2]

Classification hierarchy of the datasets. The *Safeguarding reports* dataset consists of 5 overall classes and 34 subclasses while *20 Newsgroups* contain 20 subclasses split between 6 overall classes. The default classification task for the *Toxic comments* dataset consisted of predicting different levels of toxicity in user comments on Wikipedia. Originally, the dataset contained 8 classes: 'non-toxic', 'toxic', 'severe toxic', 'obscene', 'threat', and 'identity hate' where the toxic-related labels can overlap. For providing a hierarchical classification structure, we combine all toxicity-related labels under the label 'toxic'. In this way, we create a class hierarchy with two overall classes - 'toxic' and 'non-toxic' where the 'toxic' class is overarching 6 subclasses.

Dataset	Domain	Av len	Class	Subc	Test
20 Newsgroups	Newsgroups	285	6	20	6,728
Toxic comments	Wikipedia	46	2	5	63,978
Safeguarding	Social reports	18	5	34	284

Table 1: Overview of the text classification datasets: Average number of tokens per instance (*Av len*), number of classes (*Class*), number of subclasses (*Subc*) and number of test instances (*Test*).

Filtering training data. We focus on few-shot scenarios where the dataset is balanced. We start experiments with 5 and 10 instances per label, extracted randomly from the original data ('base' instances), with at least one instance per subclass. Then, we add 5, 10, and 20 artificially generated instances to the 'base' instances ('add' instances) in order to evaluate the effect of methods over different sized training data (consisting of both original and artificially generated samples).

Domain data. In addition to the datasets with a limited amount of labels, we also leverage domain-specific corpora (in the form of the original training sets for each dataset, without making use of the labels) with two purposes: (1) analyzing the effect on GPT-2 fine-tuned on more data for generating new instances, and (2) recreating a usual scenario in practice, which is having a relatively large unlabeled corpus (e.g., many comments in the toxicity dataset, safeguarding reports which are open domain or a large number of newsgroups) but a small number of annotations. The corresponding domain corpora were also used by FastText (Bojanowski et al., 2017) to learn domain-specific embeddings.

4.2 Text Generation

As mentioned in Section 3, we use the GPT-2 language model (Radford et al., 2019) for generating additional training instances. We fine-tuned the GPT-2 model using the GPT-2 Hugging Face default transformers implementation (Wolf et al., 2019). In addition to the pre-trained general-

[1] Code to reproduce our experiments is included in the supplementary materials.

[2] Class hierarchy of the datasets and statistics for the unmodified datasets is presented in the appendix.

domain model, we fine-tune GPT-2 in each training set as well as per label using 4 epochs and default settings. For generating additional training sequences we use the sampling method of Holtzman et al. (2019).

4.3 Classification

As mentioned in Section 3.3, we use fastText[3] as our text classifier (Joulin et al., 2017, FT) where we use 'softmax', 2 grams, and domain-trained word embeddings. In order to learn domain-specific word embedding models we used the corresponding training sets for each dataset by using fastText's skipgram model (Bojanowski et al., 2017). We use fastText word embeddings rather than other word embedding models as they tend to deal with OOV words better than Glove and word2vec approaches. Also, fastText embeddings are the default using the fastText classifier.

Evaluation metrics. We report results based on the standard micro- and macro- averaged F1 (Yang, 1999).

4.4 Data Augmentation Baselines

For our baselines, we employ synonym, word embedding and language model based strategies for word replacement, and back-translation for sentence replacement (see Section 2 for more details on DA techniques). As implementations, we rely on *TextAttack* (Morris et al., 2020) for the synonym and word embedding approaches, and *nlpaug* (Ma, 2019) for the language model and back-translation. We follow the default configurations for both libraries, where WordNet (Miller, 1998) is used as a thesaurus for synonym replacement, BERT (Devlin et al., 2019) (*bert-uncased-large*) as the language model, and Transformer NMT models (Vaswani et al., 2017) trained over WMT19 English/Germany corpus for back-translation.

4.5 Case Study with Human Experts

We conducted the case study and consequently the expert-guided seed selection strategy only for the safeguarding domain where the class framework is created by subject-matter experts. While the main experiments are performed at the default sentence level, for this case study we include experiments at the passage level[4] in order to evaluate performance of text generation methods for generating both short and long sequences.

For the purpose of the experiments, we randomly selected two samples from the original data, one consisting of sentences ('sentence sample') and another one consisting of passages ('passages sample'). Each sample contained 20 instances per label or 100 instances in total. The 'sentence sample' and the 'passage sample' were distributed among two experts. Participants were asked for each sentence/passage to choose whether it is a *good* or *bad* representative of the class, or to indicate whether they are unsure. We use only a sample of the original data and involve two experts in order to evaluate whether expert-guided seed selection strategy work in a real case scenario in which the selection process is time- and cost- consuming for larger datasets. The results from the experiments show that experts selected more than 10 instances per theme for both samples as 'good representatives'. To select 10 and 5 seeds from the 'good representatives' we use random selection and max-noun selection strategies. We have included detailed information on how the study was conducted and results in the appendix.

5 Results and Analysis

The aims of our analysis is (1) to identify the most suitable method for fine-tuning GPT-2 model to ensure generating higher quality training data (see Section 5.1), and (2) to understand whether and which seed selection strategies are beneficial for improving DA methods (see Section 5.2).

Results. The main results of our experiments are displayed in Table 2 where we compare performance of DA methods for all three datasets. In Table 3 we present results related to the expert-guided seed selection strategy for the safeguarding reports at the sentence and passage levels.

5.1 *Can GPT-based Data Augmentation Help Few-Shot Text Classification?*

The results in Table 2 indeed confirm the benefits of GPT-based data augmentation. Comparing different methods for fine-tuning GPT-2 models for DA, the classification results show that GPT-2

[3]In our initial experiments we also experimented with BERT (Devlin et al., 2019) and found a similar trend but worse results overall (a small comparison between fastText and a BERT-based classifier is available in the appendix). We also provide classification results based on fastText trained on the entire non-augmented training sets in the appendix.

[4]Passages in the safeguarding reports are a list of a few sentences which could be viewed as short paragraphs. The labels for the classification remain unchanged.

	DA type	Tuning type	DA method	Micro-F1				Macro-F1			
				5base		10base		5base		10base	
				+5add	+10add	+10add	+20add	+5add	+10add	+10add	+20add
20 Newsgroups	None	-	-	.509		.578		.481		.567	
	TG (GPT2)	gen	random	.539	.536	.572	.555	.519	.519	.564	.548
		dom	random	.526	.502	.548	.539	.511	.485	.534	.526
		label	random	**.609***	**.602***	**.627***	**.637***	**.591***	**.587***	.615	.627
			nouns	.569	.549	.599	.576	.552	.533	.583	.562
			subclass	.563	.585	.624	.632	.549	.571	**.620***	**.628***
	WR	-	BERT	.519	.516	.567	.571	.511	.505	.554	.556
		-	embeddings	.556	.540	.556	.552	.534	.516	.544	.539
		-	synonyms	.517	.508	.554	.549	.502	.493	.542	.537
	SR	-	translation	.529	.525	.559	.563	.515	.509	.549	.552
Original data (upperbound)				.601	.641	.648	.654	.589	.624	.633	.639
Toxic comments	None	-	-	.423		.442		.423		.442	
	TG (GPT2)	gen	random	.447	.424	.405	.423	.447	.424	.405	.423
		dom	random	.401	.417	.369	.343	.401	.417	.369	.343
		label	random	**.453***	**.452***	.453	.442	**.453***	**.452***	.453	.442
			nouns	.417	.399	**.502***	**.461***	.417	.399	**.502***	**.461***
			subclass	.427	.440	.419	.421	.427	.440	.419	.421
	WR	-	BERT	.447	.443	.426	.422	.447	.443	.426	.422
		-	embeddings	.441	.441	.432	.432	.441	.441	.432	.432
		-	synonyms	.423	.411	.433	.429	.423	.411	.433	.429
	SR	-	translation	.446	-	.436	-	.446	-	.436	-
Original data (upperbound)				.442	.435	.448	.463	.442	.435	.448	.463
Safeguarding (sentences)	None	-	-	.242		.316		.193		.282	
	TG (GPT2)	gen	random	.294	.326	.291	.298	.212	.235	.252	.251
		dom	random	.298	.326	.291	.302	.214	.236	.252	.250
		label	random	.295	.326	.291	.302	.213	.235	.251	.252
			nouns	**.358***	**.368***	.361	**.389***	**.285***	**.302***	.327	**.358***
			subclass	.330	.351	**.372***	.329	.281	.301	**.338***	.290
	WR	-	BERT	.249	.284	.319	.315	.245	.274	.278	.274
		-	embeddings	.242	.280	.316	.319	.226	.259	.276	.283
		-	synonyms	.256	.266	.319	.326	.241	.256	.281	.288
	SR	-	translation	.287	.294	.336	.329	.257	.263	.296	.291
Original data (upperbound)				.368	.452	.432	.453	.332	.386	.386	.389

Table 2: FasText classification results based on Micro-F1 and Macro-F1. Text generation is based on GPT-2, where 'gen' refers to the pre-trained general-domain model, 'dom' refers to the same model fine-tuned on domain data, and 'label', fine-tuned per label. Data is split using 5 or 10 'base' instances per label plus additional 5, 10, or 20 'add' instances. The baselines we compare our approaches to are: word-based replacement (WR) and sentence-based replacement (SR).* – Best performing DA methods based on GPT-2 fine-tuned per label lead to statistically significant differences over non-augmented classification (*'None'*) based on t-test results where $p_{value} < 0.05$. The upperbound refers to the total number of training instances in each column, taken from the original training sets.

fine-tuned per label lead to better results for all three datasets, compared to the pre-trained model or GPT-2 fine-tuned on the entire dataset. Surprisingly, especially for the more generic datasets such as 20 Newsgroups and Toxic comments, the pre-trained GPT-2 model outperforms GPT-2 fine-tuned on the entire dataset. For instance, for 20 Newsgroups, the pre-trained model for '5+5' has a micro-F1 score of 0.539, while for the same setting the fine-tuned model on the domain has a micro-F1 of 0.526. The main reason for this behaviour can be found in that the fine-tuned model without using label-preservation techniques leads to label-distortions which add noise in the generated dataset.[5]

The results for the safeguarding reports at the passage level (see Table 3) show a similar trend where the pre-trained model outperforms the model fine-tuned on the entire dataset for all settings except for '5+5'. This is not the case, however, at the sentence-level where the model fine-tuned on the entire dataset performs very similarly to the model fine-tuned per label. In general, the results clearly suggest that fine-tuning the GPT-2 model on smaller but labelled data works better for classification than fine-tuning it on a larger unlabelled corpus, especially in settings with longer input sequences.

Statistical significance tests. We used t-test (Student, 1908) to measure whether TG-based

[5] We include automatically-generated samples of the different GPT-2 strategies and the baselines in the appendix and supplementary data.

	DA type	Tuning type	DA method	Micro-F1				Macro-F1			
				5base		10base		5base		10base	
				+5add	+10add	+10add	+20add	+5add	+10add	+10add	+20add
Passages	None	-	-	.326		.326		.299		.300	
	TG (GPT2)	gen	random	.298	.305	.382	.358	.254	.264	.335	.330
		dom	random	.333	.288	.323	.309	.276	.246	.287	.267
		label	random	.316	.302	.347	.326	.278	.266	.309	.287
			expert-random	**.404***	.386	.393	**.407***	**.358***	.349	.342	.352
			expert-nouns	.389	**.435***	**.410***	**.407***	.335	**.382***	**.351***	**.366***
Sentences	None	-	-	.242		.316		.193		.282	
	TG (GPT2)	gen	random	.294	.326	.291	.298	.212	.235	.252	.251
		dom	random	.298	.326	.291	.302	.214	.236	.252	.250
		label	random	.295	.326	.291	.302	.213	.235	.251	.252
			expert-random	**.337***	**.375***	**.361***	**.414***	**.298***	**.336***	**.340***	**.379***
			expert-nouns	.291	.298	.354	.375	.274	.276	.332	.351

Table 3: FasText classification results Micro-F1 and Macro-F1 from our case study with human experts in the safeguaring reports dataset. Text generation is based on GPT-2, where 'gen' refers to the pre-trained general-domain model, 'dom' refers to the same model fine-tuned on domain data, and 'label', fine-tuned per label. Data is split using 5 or 10 'base' instances per label plus additional 5, 10, or 20 'add' instances, 'pass' refers to passages and 'sent' refers to sentences. Best performing DA methods based on GPT-2 fine-tuned per label lead to statistically significant differences over non-augmented classification (*'None'*) based on t-test results where p_{value} < 0.05.

DA give a significant improvement over the non-augmented classifiers. In particular, we compared the best performing techniques, which are all based on GPT-2 models fine-tuned per label, and the base classifier ('None' in Tables 2 and 3). We use as a threshold α = 0.05. Results showed that $p_{value} < \alpha$ for every dataset. This confirms that fine-tuning GPT-2 model with a small number of labelled instances leads to consistent (and statistically significant) improvements for all datasets.[6]

5.2 Seed Selection Strategies Comparison

Results on comparing seed selection strategies for the the more generic datasets (20 Newsgroups and Toxic comments) showed that random selection is sufficient for improving classification performance over baselines, especially for smaller amount of seeds (see Figure 2). However, when larger number of seeds are used ('10 base') and more data is generated from these seeds using a selection strategy help improve classification performance. For instance, results for the toxic comments showed that for 10 base instances the max nouns-based strategy outperforms random selection with around 0.5 improvement in F1-measure with 5 additional instances and 0.2 improvement in F1-measure with 10 additional instances. In contrast, for the safeguarding reports (see Figure 3) both seed selection strategies (noun-guided and subclass-guided selection) lead to larger improvements over random selection even for a small number of seed samples.

[6]We include full results and t-test details in the appendix.

6 Conclusions and Future Work

In this paper, we presented data augmentation methods using text generation techniques and seed selection strategies for improving the quality of generated artificial sequences and subsequently classifier's performance in few-shot settings. In particular, we proposed four seed selection strategies for selecting class representative samples from the original data used to generate higher quality artificial instances. These are: random selection, subclass-guided selection, max nouns-guided selection, and expert-guided selection. Further, we analysed the effect of three fine-tuning strategies for GPT-2 on the quality of generated data used for classification.

Our results showed that GPT-2 fine-tuned per label, even using only handful of instances, leads to consistent classification improvements, and is shown to outperform competitive baselines and the same GPT-2 model fine-tuned on the entire dataset. This highlights the importance of label preservation techniques in the performance of TG-based DA methods, especially for generating longer sequences (such as passages or full documents). As part of our case study in safeguarding reports, seed selection strategies proved to be highly beneficial in this specialized domain, especially when experts are involved in the selection of class-indicative instances. For the other domains, which are also more similar to the datasets used to train GPT-2 (Newsgroups and Wikipedia), seed selection strategies do not lead to consistent improvements over a simple random selection for small number of seeds.

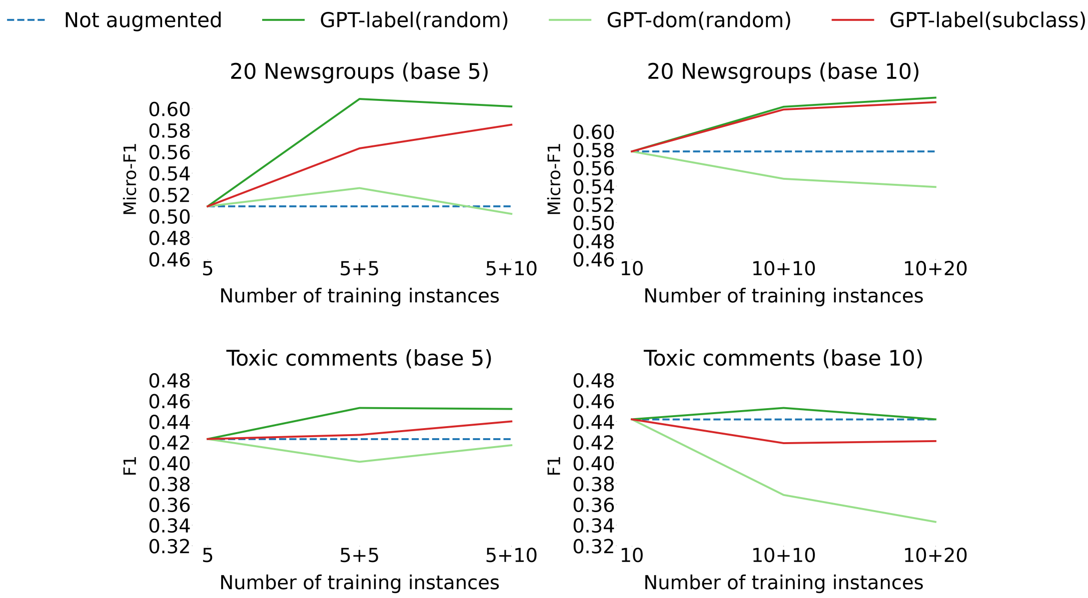

Figure 2: Micro-F1 results with 5 and 10 'base' instances per label for 20 Newsgroups and Toxic comments.

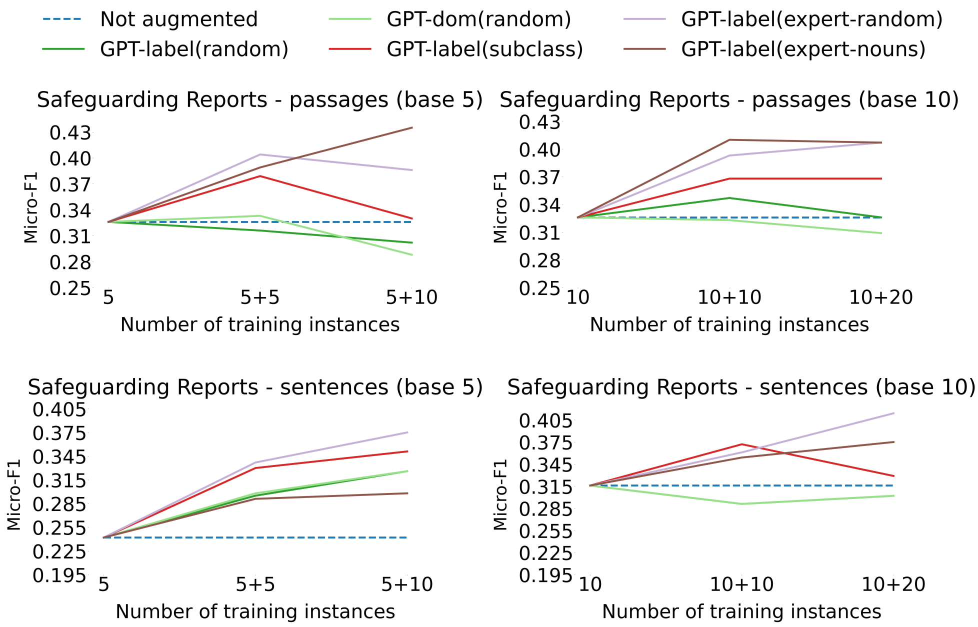

Figure 3: Micro-F1 results with 5 and 10 'base' instances per label for the Safeguarding reports dataset at the passage and sentence levels.

However, for larger numbers of seeds, strategies do help performance. In the future, we will delve further into the role of the seed selection process over the performance of DA methods considering higher number of generated additional sequences. We will also investigate the optimal number of generated instances, as our results showed a certain degradation of performance when more instances are generated. Moreover, given the positive results from the expert guided generation, we plan on exploring more methods involving human expertise into the seed selection process. Finally, our analysis could be extended to more classification tasks and domains.

References

Amanuel Alambo, Cori Lohstroh, Erik Madaus, Swati Padhee, Brandy Foster, Tanvi Banerjee, Krishnaprasad Thirunarayan, and Michael Raymer. 2020. Topic-centric unsupervised multi-document summarization of scientific and news articles. *arXiv preprint arXiv:2011.08072*.

Zuhair Ali. 2019. Text classification based on fuzzy radial basis function. *Iraqi Journal for Computers and Informatics*, 45(1):11–14.

Ali Amin-Nejad, Julia Ive, and Sumithra Velupillai. 2020. Exploring transformer text generation for medical dataset augmentation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4699–4708.

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

Ashutosh Baheti, Alan Ritter, and Kevin Small. 2020. Fluent response generation for conversational question answering. *arXiv preprint arXiv:2005.10464*.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Aleksandra Edwards, Jose Camacho-Collados, Hélène de Ribaupierre, and Alun Preece. 2020. Go simple and pre-train on domain-specific corpora: On the role of training data for text classification. In *Proceedings of COLING*.

Aleksandra Edwards, Alun Preece, and Helene De Ribaupierre. 2019. Knowledge extraction from a small corpus of unstructured safeguarding reports. In *European Semantic Web Conference*, pages 38–42, Portoróź, Slovenia. Springer.

Aleksandra Edwards, David Rogers, Jose Camacho-Collados, Hélène de Ribaupierre, and Alun Preece. 2021. Predicting themes within complex unstructured texts: A case study on safeguarding reports. In *Proceedings of the ESWC Workshop Deep Learning meets Ontologies and Natural Language Processing*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

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.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*.

Brihi Joshi, Neil Shah, Francesco Barbieri, and Leonardo Neves. 2020. The devil is in the details: Evaluating limitations of transformer-based methods for granular tasks. In *Proceedings of COLING*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Virapat Kieuvongngam, Bowen Tan, and Yiming Niu. 2020. Automatic text summarization of covid-19 medical research articles using bert and gpt-2. *arXiv preprint arXiv:2006.01997*.

Tassilo Klein and Moin Nabi. 2019. Learning to answer by learning to ask: Getting the best of gpt-2 and bert worlds. *arXiv preprint arXiv:1911.02365*.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*.

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

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339, Tahoe City, California.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Shuai Liu and Xiaojun Huang. 2019. A chinese question answering system based on gpt. In *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, pages 533–537. IEEE.

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

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

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp.

Yannis Papanikolaou and Andrea Pierleoni. 2019. Data augmented relation extraction (dare) with gpt-2. *Neuropharmacology*.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2019. Green ai. *arXiv preprint arXiv:1907.10597*.

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

Rushdi Shams. 2014. Semi-supervised classification for natural language processing. *arXiv preprint arXiv:1409.7612*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Student. 1908. The probable error of a mean. *Biometrika*, pages 1–25.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Rima Türker, Lei Zhang, Maria Koutraki, and Harald Sack. 2019. Knowledge-based short text categorization using entity and category embedding. In *European Semantic Web Conference*, pages 346–362. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Congcong Wang and David Lillis. 2019. Classification for crisis-related tweets leveraging word embeddings and data augmentation. In *TREC*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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.

Liqiang Xiao, Lu Wang, Hao He, and Yaohui Jin. 2020. Modeling content importance for summarization with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3606–3611.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. G-daug: Generative data augmentation for commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1008–1025.

Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1-2):69–90.

Danqing Zhang, Tao Li, Haiyang Zhang, and Bing Yin. 2020. On data augmentation for extreme multi-label classification. *arXiv preprint arXiv:2009.10778*.

Xinwei Zhang and Bin Wu. 2015. Short text classification based on feature extension using the n-gram model. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 710–716. IEEE.

Appendix

Original datasets: description and results

The full description of the original datasets is given in Table 4. Results from performing classification using unmodified datasets (using the full training data) are given in Table 5.

Classification Hierarchy of Datasets

Table 6 shows that for the 20 Newsgroups dataset there are 20 subclasses split between 6 overall classes. The Toxic comments consists of two overall classes - ‘toxic’ and ‘non-toxic’ where the ‘toxic’ class is overarching 6 subclasses. The Safeguarding reports consists of 5 overall classes and 34 subclasses.

Analysis GPT-2 models: Comparison between generated samples

In Table 7 we provide examples of generated instances per GPT model. Results showed that the fine-tuned model leads to miss-classifications for the 20 Newsgroup dataset and the Toxic comments dataset.

Expert Study

In this section, we describe the experiments we conducted in order to gather expert opinion on which are the theme representative samples for implementing the Expert-guided Seed Selection strategy. We performed this strategy only for the Safeguarding reports because the other two datasets are not collected from specialised domains and do not require expert annotators. We present the method followed for organising the study in Section and we present the overall results in terms of number of samples indicated as theme-representative in Section .

Method of conduct

We perform analysis for the Safeguarding reports on sentence and passage level therefore we conduct experiments for sentences and passages. The aim of this experiments is to gather domain expert opinion on which instances they consider theme representative. For these purposes, we randomly selected two samples from the original data, one consisting of sentences (‘sentence sample’) and the other consisting of passages (‘passages sample’). Each sample contained 20 instances per label or 100 instances in total. The ‘sentence sample’ and the ‘passage sample’ were distributed among two experts. The task consisted of the following: Participants were asked for each sentence/passage to choose whether it is a *good representative* of a theme or *bad representative* of the theme, or to leave space blank if they do not know. The experts followed standard procedures in thematic analysis for completing the task, similar to those used for annotating the safeguarding reports. Specifically, participants arrived to the final selection of the good theme representative samples through discussion. We use only a sample of the original data and involve a small number of experts in order to evaluate whether expert-guided seed selection strategy work in a real case scenario where domain experts are sparse and the selection process is time- and cost-consuming for larger datasets.

Expert Study Results

The results from the experiments, presented in Table 8, show that experts have selected more than 10 instances per theme for both samples as ‘good representatives’. To select 10 and 5 seeds from the ‘good representatives’ we use random selection and max-noun selection strategies. The later approach will not lead to significant changes when 10 instances are selected (i.e., some themes are associated with 11 ‘good’ instances), however it may lead to changes when experiments are performed with 5 instances.

Statistical significance test

To further evaluate the affect the additional data generated with GPT-2 have over the classifier’s performance, we performed a statistical test, t-test (Student, 1908), used to compare the means of two groups. It is used to determine if there is a significant difference between the means of two groups, which may be related in certain features. It is often used to determine whether a process or treatment actually has an effect on the population of interest, or whether two groups are different from one another.

We use t-test to measure whether the addition of GPT-2 generated training data does actually lead to improvements compared to non-augmented classifier. We specifically perform t-test between best performing seed selection strategy, highlighted in bold and ‘None’ row in Tables 3 and 4). Our H_0

Dataset	Domain	Task	Type	Class	Subclass	Avg tokens	# Train	# Test
Safeguarding Reports (passages)	Social science	Theme detection	Document	5	34	45	1,261	284
Safeguarding Reports (sentences)	Social science	Theme detection	Document	5	34	18	3,591	284
20 Newsgroups	Newsgroups	Topic categorization	Document	6	20	285	11,231	6,728
Toxic comments	Wikipedia	Toxic prediction	Document	2	5	46	159,571	63,978

Table 4: Description of unmodified datasets used in paper experiments

Dataset	Micro-F1	Macro-F1
20 Newsgroups	0.768	0.759
Toxic comments	0.908	0.908
Safeguarding Reports (passages)	0.463	0.404
Safeguarding Reports (sentences)	0.505	0.477

Table 5: FastText classification results for the entire datasets with with no augmentation.

is: *Generated data does not lead to overall improvements in classifier performance* and H_a: *Generated data does lead to overall improvements in classifier performance*. We use as a threshold α = 0.05. Results in Table 9 showed that $p_{value} < \alpha$ for every dataset. This confirms that augmenting approaches using seed selection strategies do lead to improvements in classifier's performance versus non-augmented classifiers.

Complete results from fastText classification comparing safeguarding reports, toxic comments, and 20 newsgroups

fastText classification results based on Macro-F1

Table 4 presents the results from fastText classifier for the safeguarding reports and the 20 newsgroups dataset based on Macro-F1

Dataset	Label	Sub-labels
Toxic comments	non-toxic	non-toxic
	toxic	mild toxic, severe toxic, obscene,threat, insult,identity hate
Newsgroups	computers	comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x
	recreational activities	rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey
	science	sci.crypt, sci.electronics, sci.med, sci.space
	forsale	misc.forsale
	politics	talk.politics.misc, talk.politics.guns, talk.politics.mideast
	religion	talk.religion.misc, alt.atheism, soc.religion.christian
Safeguarding Reports	Contact with Agencies	Health Practitioners, Contact with Third sector orgs, Educational Institutions, Contact with Social Care, Police Contact, Contact with councils or LAs
	Indicative Behaviour	Lying, Offending, Serious Threats to Life, Weapons, Emotional Abuse, Domestic Violence, Substance Misuse, Alcohol Misuse, Harassment, Self Inflicted Harm, Stalking, Controlling Behaviour, Aggression
	Indicative Circumstances	Bereavement,NFA, Homelessness or Constantly changing Address, Family Structure, Child Safeguarding, Relationship Breakdown, Debt or Financial Exploitation, Sex Work, Relationship with Children, Quality of Relationship
	Mental Health Issues	Children, Victim, Perpetrator, Suicidal Ideation
	Reflections	Reports Assessments and Conferences, Failures or Missed Opportunities

Table 6: Subclasses for the three datasets

Dataset	Label	Original Text	GPT(gen) generation	GPT(dom) generation	GPT(label) generation
20 News-groups	Computers	There are 3 alternatives: You can try to write a small C program something like: int my_int_var;my_int_var = system(command);	4. It's a bit confusing and requires very little programming knowledge.5. It uses only your current CPU and some extra memory.	I am now posting the following FAQs to a group of individuals whose answers are from people who are now either atheists, agnostics, etc.	It appears that I am having a problem in trying to use XRemoteAddInput(xmodmap) in windows.
Toxic Comments	non-toxic	think the recent editing by all has been brilliant. If I see anything that could be improved I'll speak up here. I would like to see a Christian Theosophy section also.	Thank you for your help, H, for your work on my blog.You are welcome to submit comments in this section, but please note I am not allowed to comment on other forums because they may interfere with my work.	No need to be an encyclopedic asshole. You don't see me deleting things because of something that was on my watchlist. I just wanted you to know that my name is now on my watchlist, and I will try to improve it.	I hope to see the article finished as soon as possible. But I would like to thank you for your contributions.
Safeguarding reports	Indicative Circumstances	Medical records show that she struggled for many years with depression, for which she was treated with anti-depressants, also receiving referrals for counselling. She also had issues with her physical health.	Her lawyer, Richard Gage, has asked the court to intervene if Mr. McDevitt was not criminally liable in his decision.	She attended at her GP with a range of depressive symptoms and her medication was improved. It is not known how she gained her weight but it is said that her mental health declined and she started following a diet with vitamin C, which was associated with being better at controlling her mood.	In the early 1970s she moved from Los Angeles to New York. During this time she suffered from depression and other problems, but there was little support in New York.She moved into a small apartment and was living with friends, but had to move into a house next to her apartment to support her mental health.

Table 7: Examples of generated samples using GPT-2 models, where Safeguarding Reports examples are non-verbatim due to data sensitivity.

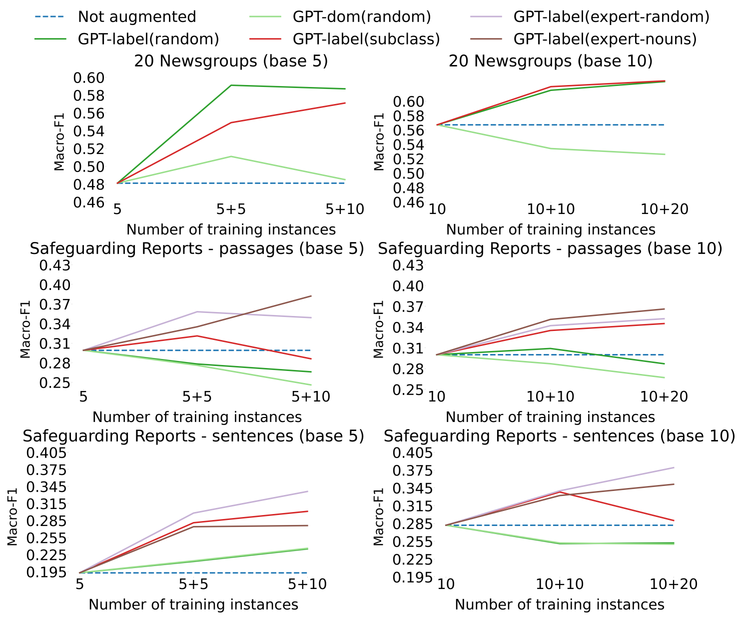

Figure 4: Macro-F1 results with 5 and 10 'base' instances per label for the Safeguarding reports dataset and 20 Newsgroups dataset

Theme	passages		sentences	
	#good rep	#bad rep	#good rep	#bad rep
Contact with Agencies	12	8	13	7
Indicative Behaviour	12	8	15	5
Indicative Circumstances	11	9	13	7
Mental Health Issues	11	9	14	6
Reflections	11	9	11	9
Total	57	43	66	34

Table 8: Results from expert study where '#good rep' refer to the number of good representative seeds that the expert selected while '#bad rep' refer to the number of samples that the expert deemed not good representatives of the themes

Dataset	p_{micro}	p_{macro}	α
20 Newsgroups	0.01	0.02	0.05
Toxic comments	0.03	0.03	0.05
Safeguarding Reports (passages)	0.0001	0.0001	0.05
Safeguarding Reports (sentences)	0.006	0.016	0.05

Table 9: T-test results - compare classification performance with no additional data and results with additional data where performance is the highest

	DA type	Tuning type	DA method	**Micro-F1**				**Macro-F1**			
				5base		10base		5base		10base	
				+5add	+10add	+10add	+20add	+5add	+10add	+10add	+20add
20 Newsgroups	None	-	-	.509		.578		.481		.567	
	TG (GPT2)	gen	random	.539	.536	.572	.555	.519	.519	.564	.548
		dom	random	.526	.502	.548	.539	.511	.485	.534	.526
		label	random	**.609***	**.602***	**.627***	**.637***	**.591***	**.587***	.615	.627
			nouns	.569	.549	.599	.576	.552	.533	.583	.562
			subclass	.563	.585	.624	.632	.549	.571	**.620***	**.628***
	WR	-	BERT	.519	.516	.567	.571	.511	.505	.554	.556
		-	embeddings	.556	.540	.556	.552	.534	.516	.544	.539
		-	synonyms	.517	.508	.554	.549	.502	.493	.542	.537
	SR	-	translation	.529	.525	.559	.563	.515	.509	.549	.552
Original data (upperbound)				.601	.641	.648	.654	.589	.624	.633	.639
Toxic comments	None	-	-	.423		.442		.423		.442	
	TG (GPT2)	gen	random	.447	.424	.405	.423	.447	.424	.405	.423
		dom	random	.401	.417	.369	.343	.401	.417	.369	.343
		label	random	**.453***	**.452***	.453	.442	**.453***	**.452***	.453	.442
			nouns	.417	.399	**.502***	**.461***	.417	.399	**.502***	**.461***
			subclass	.427	.440	.419	.421	.427	.440	.419	.421
	WR	-	BERT	.447	.443	.426	.422	.447	.443	.426	.422
		-	embeddings	.441	.441	.432	.432	.441	.441	.432	.432
		-	synonyms	.423	.411	.433	.429	.423	.411	.433	.429
	SR	-	translation	.446	-	.436	-	.446	-	.436	-
Original data (upperbound)				.442	.435	.448	.463	.442	.435	.448	.463
Safeguard (pass)	None	-	-	.326		.326		.299		.300	
	TG (GPT2)	gen	random	.298	.305	.382	.358	.254	.264	.335	.330
		dom	random	.333	.288	.323	.309	.276	.246	.287	.267
		label*	random	.316	.302	.347	.326	.278	.266	.309	.287
			nouns	.375	.337	**.375**	**.379**	.329	.281	**.338**	**.351**
			subclass	.379	.330	.368	.368	.321	.286	.335	.345
	WR	-	BERT	.287	.294	.326	.336	.282	.278	.294	.297
		-	embeddings	**.389**	**.382**	.305	.319	**.343**	**.341**	.283	.287
		-	synonyms	.277	.267	.312	.315	.256	.245	.285	.292
	SR	-	translation	.333	.336	.298	.312	.294	.301	.273	.286
Original data (upperbound)				.336	.337	.358	.368	.301	.304	.307	.320
Safeguard (sent)	None	-	-	.242		.316		.193		.282	
	TG (GPT2)	gen	random	.294	.326	.291	.298	.212	.235	.252	.251
		dom	random	.298	.326	.291	.302	.214	.236	.252	.250
		label	random	.295	.326	.291	.302	.213	.235	.251	.252
			nouns	**.358***	**.368***	.361	**.389***	**.285***	**.302***	.327	**.358***
			subclass	.330	.351	**.372***	.329	.281	.301	**.338***	.290
	WR	-	BERT	.249	.284	.319	.315	.245	.274	.278	.274
		-	embeddings	.242	.280	.316	.319	.226	.259	.276	.283
		-	synonyms	.256	.266	.319	.326	.241	.256	.281	.288
	SR	-	translation	.287	.294	.336	.329	.257	.263	.296	.291
Original data (upperbound)				.368	.452	.432	.453	.332	.386	.386	.389

Table 10: FasText classification results based on Micro-F1 and Macro-F1. Text generation is based on GPT-2, where 'gen' refers to the pre-trained general-domain model, 'dom' refers to the same model fine-tuned on domain data, and 'label', fine-tuned per label. Data is split using 5 or 10 'base' instances per label plus additional 5, 10, or 20 'add' instances, 'sent' refers to sentences. The baselines we compare our approaches to are: the word-based replacement (WR) and sentence-based replacement (SR) strategies.* – Best performing DA methods based on GPT-2 fine-tuned per label lead to statistically significant differences over non-augmented classification (*'None'*) based on t-test results where $p_{value} < 0.05$.