ContextGen: Targeted Data Generation for Low Resource Domain Specific Text Classification

Lukas Fromme
University of Stuttgart (IMS)

Jasmina Bogojeska
IBM Research Zurich

Jonas Kuhn
University of Stuttgart (IMS)

Abstract

To address the challenging low-resource non-topical text classification problems in domain specific settings we introduce ContextGen – a novel approach that uses targeted text generation with no fine tuning to augment the available small annotated dataset. It first adapts the powerful GPT-2 text generation model to generate samples relevant for the domain by using properly designed context text as input for generation. Then it assigns class labels to the newly generated samples after which they are added to the initial training set. We demonstrate the superior performance of a state-of-the-art text classifier trained with the augmented labelled dataset for four different non-topical tasks in the low resource setting, three of which are from specialized domains.

1 INTRODUCTION

Text classification is a task in Natural Language Processing (NLP), widely used across practical applications ranging from spam detection, via news categorization, sentiment analysis and question answering to dialog systems. Recent advances in deep learning research have made it possible to reach very high prediction performance for text classification tasks on general knowledge domain data when a sufficiently large labelled dataset is available. However, in specialized domains one is often confronted with complex text classification tasks for which the available annotated data are very limited. Many categorization distinctions are non-topical which makes supervised training with limited data particularly difficult.

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

Furthermore, annotation in specialized domains is often expensive as it can only be performed by domain experts. Classifying medical, contractual or incident ticket texts are few such examples. The work we present in this paper introduces a novel, data-augmentation based approach that aims to improve prediction quality for domain-specific, non-topical text classification tasks in a low-resource setting. By **non-topical** we refer to classification tasks where the samples of a given class do not reflect a certain topic but a different semantic aspect of the text. Usually, topical classification tasks try to determine the domain or subject of a sentence such as deciding if a sentence deals with a flight or a call or if a customer wants to book an appointment. In non-topical tasks one could classify for example sentiment, user intent or subjectivity. While our method can also be applied to topical classification tasks, we believe that non-topical tasks in specialized domains are generally more challenging, highly relevant in industrial application (e.g., in legal, medical or forensic domains) and have not received much attention in the scientific community. Our method, referred to as ContextGen, builds upon the powerful GPT-2 text generation model (Radford et al. (2019)) to enrich the sparse training dataset by producing novel examples relevant for a given target classification problem. The synthetic examples are generated without any GPT-2 fine-tuning which is impractical in limited resource settings. Instead, we introduce and evaluate two approaches that use cues from the texts of the available labelled samples as input to the generation model. This way we provide appropriate context to ensure the newly generated samples are relevant for the target domain. There are two alternatives for assigning a label to the generated samples, namely, the label is identical to the label of the sample that was used to extract the cue for generation, or the label is assigned using a majority vote from an ensemble of text classifiers trained on the available training data. The main contributions of this work are the following: (1) We contribute ContextGen, a novel augmentation-based method to address domain-specific, non-topical text classification tasks when limited amount of la-

belled data are available (see Appendix for the full Code). (2) We present evaluation results showing the performance of our approach on several datasets. (3) We analyse the newly generated samples.
Our focus is on binary classification tasks which often occur in the non-topical text classification scenario but our findings are equally relevant for multiclass classification and our approaches can be easily applied there too. The rest of the paper is organized as follows. Section 2 gives an overview of related work. In Section 3 we provide the detailed methods for our approach, Section 4 describes the experiments and Section 5 discusses the experimental results. Section 6 concludes our paper.

2 RELATED WORK

Feng et al. (2021) define three groups of augmentation techniques for NLP, namely: Rule-Based Techniques, Example Interpolation Techniques and Model-Based Techniques. ContextGen falls into the latter category of Model-Based Techniques which have become more and more relevant due to the rise of powerful pre-trained language models (LMs). One such approach, detailed in Kobayashi (2018) substitutes selected words in the training sentences with corresponding synonyms which also keeps the meaning of the original sentence intact. Synonyms are selected using a language model (see also Wu et al. (2019)), but can also be determined using lexical resources (Jungiewicz and Smywinski-Pohl (2019)). Other model-based approaches rely on generative models to create new sentences using the existing training data as a basis. One idea is to use the original sentence as a skeleton and fill in blanks using MaskGan (Fedus et al. (2018)). A new LM is trained and then used to generate new sequences for masked parts of the sentence using a Generative Adversarial Network architecture.
Instead of training from scratch, pre-trained language models such as GPT-2 (Radford et al. (2019)) are capable of generating large amounts of human-quality text. However, controlling the content of the generated text is challenging. Conditional language generation (Keskar et al. (2019)) addresses this problem by fine-tuning the generative model on data where each sentence is prepended by a keyword. Giving the keyword as input to GPT-2 enables generation of texts similar to the ones in the training data which share the keyword. This technique is employed for data augmentation of a classification task in Anaby-Tavor et al. (2019). Here, the classification labels are used as the keywords for conditional generation, resulting in a model that produces text pertaining to a certain class of the dataset. In general, these model-based techniques require either training a completely new LM or some sort of fine-tuning step, which means that the pre-trained LM is trained again for several iterations on the available data before using it for inference. This is a necessary step to ensure that the text generated by the LM output is relevant for the current dataset. For ContextGen, however, we focus on domain-specific datasets where a small amount of data are available. This can make the fine-tuning problematic since the LM does not see enough data to properly generalize. As such, fine-tuning of a language model is impractical in low resource domain-specific settings. The classical approach to data augmentation is using rule-based methods. Easy Data Augmentation (EDA) (Wei and Zou (2019)) proposes the use of a set of simple editing techniques - including synonym substitution - as a baseline for future augmentation experiments. Warping sentences in this manner mirrors the techniques used in image classification (Shorten and Khoshgoftaar (2019)), which geometrically transform images, for example by mirroring or flipping. Finally, the back-translation method first translates sentences into a different language and then translates them back to the original language using machine translation (Yu et al. (2018)). Due to the different characteristics of the two languages, this approach also creates edited versions of the original sentences though with limited variability. Furthermore, translation models are not trained on specialized domains, which impacts the translation quality in such cases.

3 CONTEXTGEN METHOD

To address the small amount of labelled data available in low resource tasks, the ContextGen method uses data augmentation by first adapting the GPT-2 [1] text generation model to generate synthetic samples relevant for the target domain and then assigning labels to the new samples. In the following, we provide the details of our approach.
The small training set sizes in a domain specific setting hamper successful fine-tuning of text generation models. In order to generate synthetic samples relevant for the target domain, we use properly selected text from the samples in the training set as input to the GPT-2 model without any fine-tuning. Intuitively, the textual cues give the necessary context that enables the language model to generate samples relevant for the target domain. The ContextGen method is illustrated in Algorithm 1.

Our augmentation algorithm works by first sampling a subset D_{sub} from the original training data D. The samples in D_{sub} are used to create proper contexts

[1] 'gpt2-medium' model available at https://huggingface.co/transformers/pretrained_models.html

Algorithm 1 ContextGen: Generating Context-Dependent sentences without fine-tuning GPT-2. Dataset D consists of label-sentence pairs *(l, s)*

```
D_sub = random subset of D
for Every (l, s) in D_sub do
    Build context c by concatenating s and selected set of words from s
    Generate s* using c as the input to GPT-2
    if using ensemble voting then
        Choose new label l* with ensemble majority vote
    else
        Set new label l* equal to label l
    end if
    add (l*, s*) to D
end for
```

for the GPT-2 generation. Finally, appropriate class labels are assigned to the newly generated sentences and they are used to augment the initial small, labelled dataset.

3.1 Text Generation

We introduce two different approaches to create the context string used as input for the text generation model.

Sentence-beginning context. We build context c by concatenating a target sentence s with all words of s up to (and including) the first noun phrase (NP). Using the NP we make sure to give a meaningful part of the sentence as input to the generation. Examples can be found in Table 1. **Attention-target context.** First, we fine-tune a BERT classifier (Devlin et al. (2018))[2] using the available labelled data. Then, we extract all NPs from a target sentence s, rank them by how strongly they are weighted by the classifiers attention. This reflects their importance for the classification task and thus we select the highest ranked NP to use in the context text. More specifically, for each layer in the BERT model, we check which head is targeting which word with the highest attention and calculate word ranking based on the times each of them is targeted across all attention layers. We select the NP from the target sentence that contains the highest ranked word in the word attention ranking. In case of a tie when more than one NP contains words with identical rank, we use the NP which occurs earlier in the sentence. The context c is created by appending the selected NP to the sentence s. This way, GPT-2 is able to generate information pertaining to both the domain given in the target sentence s as well as the entity described in the NP most important for the classification task. An example result using attention-target context can be found in Table 1.

[2]Using the pre-trained *bert-base-uncased* model available on https://huggingface.co/models.

3.2 Class Label Assignment

To augment the labelled dataset with the newly generated, domain-specific samples we need to assign them a class label. We have two alternatives for the label assignment.

The first one is to use the same class label as the one of the sample used to generate the context input for the generation. While the text generation model is able to adapt to the domain of a given context, we do not control the exact contents of generated sentences and have no guarantees that the label will be preserved. Thus, the second alternative is to assign class labels by utilizing an ensemble approach where several pre-trained BERT classifiers (Devlin et al. (2018)) are fine-tuned on the corresponding non-augmented data with different random initialization. We use BERT due to the excellent performance of pre-trained transformer models for text classification tasks. More specifically, every classifier in the ensemble predicts a label for the newly generated sentence s^*. The new label for s^* is then chosen by majority vote [3](i.e., we assign the label which was predicted the most times by the ensemble). We define the majority as having more than 50% of total votes.

4 EXPERIMENTAL SETTING

4.1 Datasets

For our experiments we consider and adapt four different public datasets, three of them from specialized domains, along with challenging, non-topical binary text classification tasks. In the following we provide more details for each of them:

DISONLY (Rosario and Hearst (2004)) The original corpus contains sentences taken from abstracts of medical journals. The sentences have labelled spans that detail information such as *Disease* or *Treatment*. We transform the annotations into a binary classification problem by separating sentences that only contain information about diseases from the rest. That means, that a sentence containing both information about a disease as well as something else (e.g., the patient) will not belong to the target class. To clarify: we do not use the span information from the original corpus. Ev-

[3]We also experimented with other voting techniques, such as leaving out the worst models or increasing the threshold for majority to 75% instead of 50%. However, we found that none of our modifications influenced the results in a significant way.

Context (original sentence)	Generated sentence
Of 75 women, drug users (51 %) were more likely to say that they would defer initiating prenatal care (P = 0.03) and to minimize the risk of drug or alcohol use to the fetus (P = 0.04) .	**Of 75 women,** 85 were diagnosed with acne after taking anti-depressants (21/55) and 45 were found to be "normal" (9/55). (These data were obtained from the study by Karamala et al.
(2) the lessee may demand from the lessor, at the earliest four months before the expiry of the time limit, that the lessor informs him within one month whether the reason for the time limit still exists.	**(2) the lessee** may not use the electronic communication devices of the landlord in the manner described in paragraph (1) if the electronic communication device is a wireless device
The genetically significant dose from roentgen examinations in Finland in 1963.	**roentgen examinations** are intended to assist in the diagnosis and treatment of the following conditions

Table 1: Examples of well-formed ContextGen generations from several datasets. The generation is performed by giving the context, as well as the words marked in bold to a pre-trained GPT-2 (Radford et al. (2019)) model. The first two examples show results for the sentence-beginning context, the last example shows a generation with the attention-target context.

ery sentence has exactly one label. The final dataset consists of 3445 sentences, 607 of which are labelled with the target class.

RIGHT[4] We use a small corpus of German sentences from tenancy law that are annotated with various labels. We translate the sentences to English [5] and create a class of sentences that contain or detail a person's right. The final corpus contains 526 sentences, 200 of which are annotated as containing a person's right.

TOS (Lippi et al. (2019)) This corpus analyses sentences from terms-of-service agreements in order to find potentially unfair sentences. We use this information and treat it as a binary classification problem. Of the 9415 sentences in the dataset, 1032 are labelled as potentially unfair.

SUBJECTIVITY (Pang and Lee (2004)) Using sentences from both movie reviews and plot summaries, this dataset has two classes: Subjective and Objective. We use the dataset as it is provided without making any changes. The dataset contains 2000 sentences. 1000 sentences are labelled as subjective, the other half is labelled as objective. Note that while we have a non-topical classification problem this dataset pertains to the common knowledge domain.

We split all datasets randomly into 60% training set, 20% test set and 20% validation set.

4.2 Baselines

We compare our augmentation technique against three baseline approaches:

No Augmentation The text classification model is trained on the original, non-augmented dataset.

[4]https://github.com/sebischair/Legal-Sentence-Classification-Datasets-and-Models

[5]Translation was done in a semi-automatic manner, using a Neural Machine Translation model and manually checking for errors afterwards.

$$F1(l^*, l) = 2 * \frac{precision(l^*,l) * recall(l^*,l)}{precision(l^*,l) + recall(l^*,l)}$$

Figure 1: Formula for calculating the F1 score given a sequence of predicted labels l^* and a sequence of annotated labels l.

EDA Wei and Zou (2019) warp existing sentences by swapping words, deleting single words or inserting (or substituting) synonyms. [6] The authors show improvements in classification accuracy on several benchmark datasets. The authors also demonstrate that their method preserves the original class label, thus requiring no extra mechanism to determine the label of generated sentences. *EDA* was tested on both multi-label and binary classification tasks.

Lambada Anaby-Tavor et al. (2019) also use the GPT-2 language model to generate sentences that augment the original labelled dataset. However, the authors fine-tune the language model with the non-augmented dataset in order to fit class labels to sentence generation which are further filtered using the confidence score of a classifier trained on the original dataset. Lambada improves classification accuracy on several benchmark multi-label topical classification tasks.

We also consider a number of additional baselines to test individual components of ContextGen (see Section 3). **Random NP** uses a random noun phrase instead of the attention target. **Random Gen** generates from GPT-2 without context. **RNN** trains a recurrent neural network language model on the full dataset from scratch (i.e., without fine-tuned word embeddings) and then generates sentences using the contexts of the ContextGen approach. **NP only** augments the training data only with the attention target noun phrase without any generated content.

[6]Synonyms are determined with WordNet (Miller (1995))

4.3 Setup

We train a BERT model for binary text classification using the transformers library from Huggingface [7] for 30 epochs with a batch size of 16 and early stopping given the performance on the development set. As most of the datasets have imbalanced classes, we report the F1 score (see Figure 1) for all results. Our parameter selection is also based on the F1 score.
Furthermore, we train 10 BERT classifiers per experiment each with different random initialization. We select the model that achieves the best F1 score on the development set. The label voting strategy utilized for class assignment is based on an ensemble of 10 BERT classifiers (see Section 3.2).
In order to investigate the effect of augmentation on particularly small training datasets, we run experiments for randomly sampled subsets of each dataset. We sample subsets of sizes 50, 100, 200, 500 and 1000 respectively. The exception is the RIGHT dataset where only 314 samples are available. We further sample each subset size three times using a different random seed each time. All results reported are the averages of these three runs on the different dataset splits. When augmenting, we randomly select a percentage of sentences from the current training set to use as context. The sampled sentences have the same class balance as the original set (at least one sentence per class). We run experiments for four different augmentation percentages: 10%, 50%, 100%, 200%. This means that, at most, we triple the size of the training set when augmenting. All experiments were run on a single Nvidia GTX 1080 GPU. For the final results reported on the test set, we choose the best models based on their F1 scores on the development set.

5 RESULTS AND DISCUSSION

We report the performance of the best models on each test set selected based of the models on the corresponding development-set in Table 2 and illustrated against the non-augmented baseline in Figure 2.
On the RIGHT dataset, the sentence beginning context does not improve upon the baselines for training set sizes of 50 and 100 sentences but matches or improves the baseline F1 scores for training sets of 200 and 314 sentences. The attention-target context shows higher F1 for the training set sizes of 50, 100 and 314 sentences than the sentence-beginning context while also improving on the baselines performance. Similarly, results on the DISONLY dataset show improvements when using the attention-target context for training sets of 50 and 100 sentences while the sentence-beginning context performs worse than the baselines on these dataset sizes. For sizes 200 and above, both contexts reliably improve on the baselines performance with F1 gains of up to 0.13 when using sentence-beginning context with training set sizes of 500 sentences. Finally, ContextGen achieves high performance gains of up to 0.16 F1 on the TOS dataset when using attention-target context on training sets comprising 100 sentences. Both context types consistently improve upon or match the baseline approaches on the other subsets with the exception of sentence-beginning context with training sets of 200 sentences which is slightly below the non-augmented baseline. ContextGen delivers comparable results on the SUBJECTIVITY dataset, as the non-augmented baseline score is very high with an F1 of around 0.9 even on 50 sentences. 5 shows the best performing ContextGen models which do not significantly improve upon the non-augmented baseline. As such, we exclude the SUBJECTIVITY dataset from further investigation. Full results for the dataset can be found in the Appendix in Table 4.

5.1 Effect of ContextGen Parameters

In order to gain a deeper understanding of the effect of ContextGen parameters, we show an overview of test F1 scores across a grid of ContextGen configurations in Table 3. We vary the type of context, whether we use label voting and the number of augmented sentences as described in Section 4.3. The results are averaged over the 10 different model initializations. On the RIGHT dataset, when using training set sizes of 50 and 100 sentences, the attention-target context yields significant improvements of 0.05 to 0.07 F1 over the baselines. For training sets of 200 and 314 sentences, the sentence-beginning context outperforms the baselines on several configurations, yielding improvements of up to 0.1 when augmenting 200% of the data. For the DISONLY medical data, we observe significant improvements over the baselines for 50 and 100 sentences in select configurations only, mostly when using the attention-target context. When using training sets comprising 200 or more sentences, ContextGen consistently outperforms the baselines with both context types. Notably we see the best overall result of 0.72 F1 when using the sentence-beginning context without label voting for training sets of 1000 sentences. Looking at the results of the TOS dataset when using 50 labelled sentences one can observe that the attention-target context with voting manages to achieve an improvement of 0.08 F1 (100% augmented, voting) while many of the sentence-beginning configurations are worse than the non-augmented baseline. Starting at training sets with 500 sentences, the sentence-beginning context outperforms the baselines

[7] Using the 'bert-base-uncased' model. Library and Model available at https://huggingface.co/transformers/

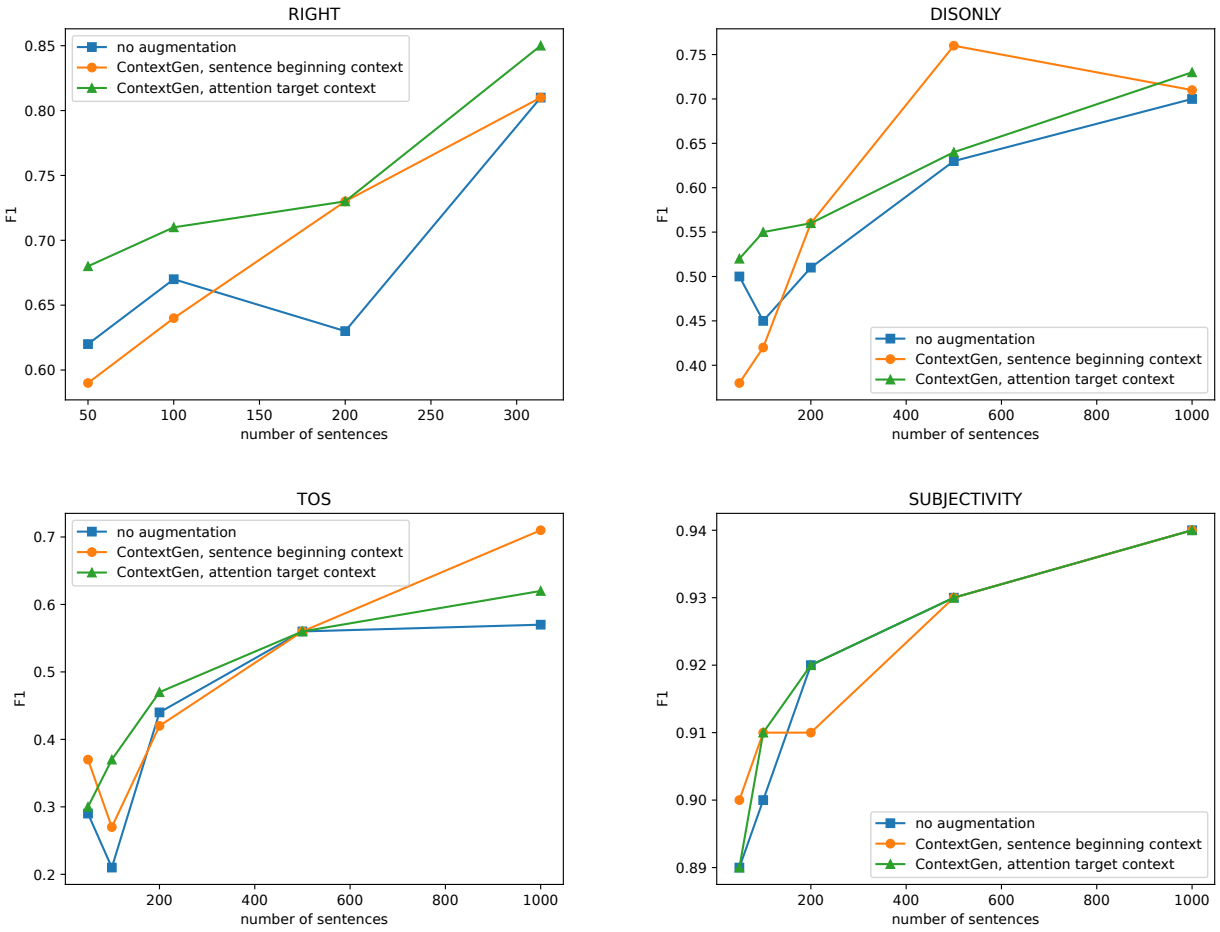

Figure 2: Performance of the best augmented models on the development set for training sets of different sizes. We sample each subset 3 times and report the average performance.

Augmentation	RIGHT				DISONLY					TOS				
# sentences	50	100	200	314	50	100	200	500	1000	50	100	200	500	1000
no augmentation	0.62	0.67	0.63	0.81	0.5	0.45	0.51	0.63	0.70	0.29	0.21	0.44	**0.56**	0.57
EDA	0.63	**0.71***	0.65	0.77	0.50	0.49	0.52	0.69	0.71	0.15	0.32	0.32	0.54	0.63
Lambada	0.40	0.65	0.71	0.71	0.33	0.32	0.48	0.62	0.71	0.22	0.25	0.25	0.50	0.61
sentence-beginning context	0.59	0.64	**0.73***	0.81	0.38	0.42	**0.56***	**0.76***	0.71	**0.37***	0.27	0.42	**0.56**	**0.71***
attention-target context	**0.68***	**0.71***	**0.73***	**0.85***	**0.52**	**0.55***	**0.56***	0.64	**0.73***	0.3	**0.37***	**0.47***	**0.56**	0.62

Table 2: Test set F1 of the best performing models out of the ensemble of 10 trained classifiers for each configuration. Models are selected based on their performance on the development set. Results are averaged over the three different dataset splits (see Section 4.3). *t-test, $p < 0.05$

while staying slightly below the attention-target context by 0.01 F1 for the best corresponding configurations. When using training sets comprising 1000 sentences, the highest improvements are observed when using the sentence-beginning context at 0.11 F1 (50% augmented without label voting and 10% augmented with label voting). Overall, we find that the additional baseline methods (random NP, random gen, rnn, NP only) generally perform worse than ContextGen. We conclude from this that the choice of noun phrase and the quality of the generation model are critical for the effectiveness of ContextGen.

Dataset	RIGHT				DISONLY					TOS				
# sentences	50	100	200	314	50	100	200	500	1000	50	100	200	500	1000
no augmentation	0.64	0.66	0.63	0.77	0.43	0.43	0.48	0.62	0.67	0.22	0.36	0.40	0.49	0.57
random NP	0.63	0.67	0.66	0.77	0.37	0.43	**0.54***	0.64	0.71	0.19	0.24	0.36	0.55	0.61
random gen	0.57	0.62	0.66	0.77	0.39	0.44	0.53	0.67	**0.72**	0.18	0.25	0.40	0.56	0.62
rnn	0.49	0.58	0.61	0.77	0.35	0.44	**0.54***	0.64	0.71	0.19	0.20	0.43	0.52	0.61
NP only	0.49	0.55	0.61	0.77	0.29	0.44	**0.54***	0.64	0.71	0.22	0.26	0.34	0.56	0.64
EDA	0.63	0.59	0.58	0.71	0.35	0.44	0.44	0.59	**0.70**	0.12	0.25	0.25	0.47	0.61
Lambada	0.28	0.59	0.62	0.73	0.17	0.15	0.37	0.56	0.69	0.12	0.12	0.12	0.44	0.55
ContextGen														
sentence-beginning context, no voting														
10% augmented	0.58	0.64	0.70	0.76	0.22	0.43	**0.54***	0.64	**0.72***	0.13	0.30	0.41	0.56	0.67
50% augmented	0.60	0.62	0.69	0.74	0.46	0.30	0.40	0.65	0.69	0.19	0.23	0.35	0.48	**0.68***
100% augmented	0.58	0.66	0.67	0.73	0.37	0.38	0.45	0.46	0.54	0.11	0.18	0.33	0.47	0.60
200% augmented	0.53	0.55	0.49	0.73	0.16	0.35	0.52	0.40	0.68	0.17	0.33	0.36	0.49	0.65
sentence-beginning context, voting														
10% augmented	0.64	0.62	0.67	0.76	0.28	0.29	0.52	**0.70***	0.70	0.15	0.29	0.34	0.50	**0.68***
50% augmented	0.66	**0.71***	0.69	**0.79***	0.28	0.29	0.52	**0.70***	0.70	0.25	0.38	0.40	0.56	0.63
100% augmented	0.61	0.69	0.63	**0.79***	0.33	0.33	**0.54***	**0.70***	0.71	0.23	0.28	0.43	0.51	0.65
200% augmented	0.64	0.71	**0.73***	0.74	0.31	0.30	0.47	0.59	0.65	0.13	0.25	0.36	0.51	0.66
attention-target context, no voting														
10% augmented	0.62	0.68	0.65	0.69	0.50	0.47	0.53	0.63	0.64	0.17	0.23	0.34	0.48	0.55
50% augmented	0.61	0.67	0.65	0.59	0.41	0.42	0.45	0.60	0.65	0.17	0.27	0.35	0.52	0.60
100% augmented	0.64	0.68	0.66	0.71	0.45	0.39	0.49	0.57	0.62	0.17	0.25	0.32	0.49	0.58
200% augmented	0.63	0.67	0.66	0.74	0.32	0.35	0.43	0.41	0.61	0.16	0.23	0.29	0.50	0.65
attention-target context, voting														
10% augmented	0.66	0.69	0.70	0.71	0.48	0.45	0.53	0.63	0.66	0.25	**0.43***	**0.45***	0.56	0.63
50% augmented	0.60	0.65	0.67	0.63	**0.51***	0.48	0.43	0.66	**0.70**	0.29	0.38	0.42	**0.57***	**0.65***
100% augmented	**0.69***	0.68	0.51	0.66	0.50	**0.52***	0.50	0.67	0.69	**0.30***	0.40	0.44	**0.57***	0.58
200% augmented	0.64	**0.71***	0.72	0.77	0.33	0.49	0.44	0.40	0.66	0.13	0.25	0.30	0.50	0.61

Table 3: Results on three domain-specific datasets for every ContextGen configuration and all the baselines approaches. *10% augmented* signifies, that we applied augmentation to a 10% sample of the training set. Best results for each subset size are marked in bold. All results are calculated using the F1 score. *t-test, $p < 0.05$

5.2 T5 Generation Model

We test a different generation model by replacing GPT-2 with the T5 (Raffel et al. (2020)) model, which has also been shown to generate high quality text. We find F1 scores to be comparable for the two models, with T5 yielding small improvements over GPT-2 on the RIGHT dataset but performing worse than GPT-2 on the DISONLY and TOS datasets. We conclude from this that both models are generally suited for this augmentation task but that T5 does not have a big advantage over the GPT-2 model. The full results are shown in Table 6 in the appendix.

5.3 Generated Content

The ultimate goal of data augmentation is to enrich the original, small training set with new information relevant for the target classification task. Figure 3 depicts two aspects of the generated sentences, namely, the variability of the new sentences compared to the context sentence, and the vocabulary extension. First, we observe that on the RIGHT dataset, using ContextGen with sentence-beginning contexts results in a large percentage (around 30%) of the generated sentences being copies of the original sentence used to provide the context for generation. In the other three datasets this is the case for only 3% of generated sentences. When using the attention-target context the variability of the generated sentences is always higher than the one obtained when using the sentence-beginning contexts. The goal of the attention-target contexts is to facilitate generation of sentences that vary greatly from the original context while still fitting into the desired domain and being relevant for the classification task. Second, we investigate potential vocabulary extension, namely, how many new words are introduced to the classifier with the augmented data. We analyze the overlap of the vocabulary between augmented and non-augmented data for both context-building methods. A low vocabulary overlap means the augmented training set contains new information (words). We can observe in Figure 3, that the use of attention-target contexts reduces vocabulary overlap by around 20% for all datasets except for DISONLY. We also analyze the similarity of augmented sentences and their respective context sentence using BLEU-score (Papineni et al. (2002)), which is an evaluation measure

typically found in machine translation that checks for multiple bigram overlaps. We calculate the BLEU-score for each augmented sentence with respect to the original context as reference and average the score over all sentences of the corresponding dataset. The results of this analysis are found in Figure 4. We see that on 3 out of the 4 datasets, for the sentence-beginning context BLEU is in the range of 0.4 to 0.6, which means that there is a significant overlap between the original and corresponding generated sentences. In addition, we observe that the attention-target context effectively reduces average BLEU by a large margin on all datasets except for the DISONLY dataset.
To summarize, with the attention-target contexts, we can effectively avoid generating sentences identical to the context sentence from the training set and even expand the initial vocabulary with new words. This is especially beneficial in domain-specific settings when the performance of the non-augmented model is not very low (like the TOS dataset).

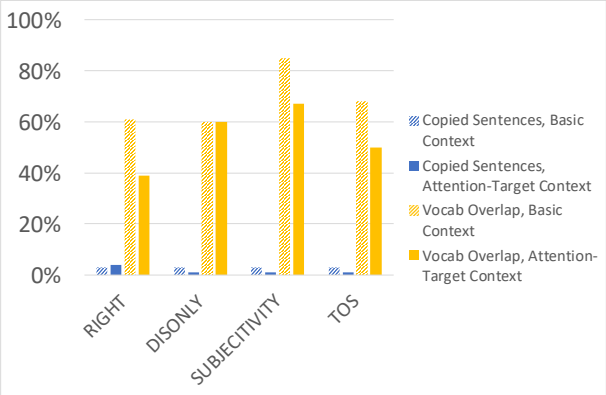

Figure 3: Copied sentences and vocabulary overlap for augmented sentences on all datasets. Dashed columns show the overlap between original sentences and sentences generated with sentence-beginning contexts. Full colour columns compare the original sentences to sentences generated with attention-target contexts.

5.4 Discussion

Our results illustrate two aspects of the proposed Method. Table 2 shows that in practice, we are generally able to improve on all baselines by selecting the best model from the development set. On the other hand, Table 3 provides more information how the ContextGen parameters change the performance on the test set. While this is not useful in practice, we can gain valuable insight into how our augmentation scheme affects the classification.
As can be observed in Table 2, our ContextGen approach outperforms most of the baseline approaches on on all dataset sizes. The chosen parameters for each model in Table 2 are listed in the Appendix (5). Furthermore, the majority of best results are obtained when using the attention-target context, especially for the small training set sizes. For example, in Table 2 we see, that when few sentences are available the sentence-beginning context yields worse performance than the baseline on the RIGHT and DISONLY datasets, while the performance is superior compared to the baselines performance when using the attention-target context. Looking into Table 3 we observe the same effect also on the TOS dataset. One possible reason is for this is, that the majority of the sentences generated with attention-target context differ from the corresponding sentence used to generate the context. Looking at the analyses in Figure 3 and 4 we see that for the RIGHT and TOS datasets, sentences augmented with sentence-beginning context have both a higher overlap in vocabulary as well as a much higher average BLEU compared to the ones generated using the attention-target context. Furthermore, generated sentences which are very similar to their corresponding original sentence with a correctly chosen label do not bring much new information to the available labelled data, while such sentences with incorrect labels even introduce noise in the training data.
On the DISONLY dataset, we observe that even though the variability of sentences is similar for both context generation types, the attention-target contexts achieve higher scores for smaller training set sizes. This shows that there is also a quality difference between the two augmentation strategies. The attention-target context focuses on nouns that are particularly important to the classifier which means that even with the same sentence variability, the augmented sentences provide useful additional information to the classifier and thus lead to higher quality classification performance when limited annotation data is available. Note however that the sentence-beginning context yields good quality performance when larger amount of training data are available.
We observe an overall low performance of *Lambada* especially for the small training set sizes (smaller than 1000 samples). The reason for this is most likely, the non-topical, domain-specific text classification setting where language-model fine tuning is impractical. While this does not definitively prove that fine-tuning on a small amount of data can not be beneficial for language generation and data augmentation, we do believe that our results illustrate the issues which will most likely extend to other model-based augmentation approaches that we did not examine in this paper.

Another interesting observation is that although a large number of best performing models are achieved when using the label voting it does not always improve the effect of augmentation. More specifically, using the label voting with either context generation approach can yield worse performance compared to using the original label for some of the small size training sets. One possible explanation is that the generated sentences are fairly similar to the corresponding original context sentences and thus the original label remains valid. As such, we can assume that many of the augmented sentences carry large parts of the original meaning of the context and therefore using the label of the original sentence is often correct. However, we can assume that this effect does not persist when dealing with more than two classes in a multi-class or even multi-label setting. Another consideration is the performance of the non-augmented model itself which greatly affects the quality of the label voting. A low F1 from the classifier would translate to a worse label-voting mechanism. In fact, we see the best voting results when the non-augmented baseline is also high, for example on the RIGHT dataset when using all 314 sentences. It should generally be noted that despite individual configurations in Table 3 showing decreased F1 when using label voting, a large number of best performing models are achieved when using it. For future experiments, we believe that looking into more ways of generating context cues from the data will be worthwhile. We show in Section 5.3 that the cue which is given to the generative model greatly affects the diversity of its output which in turn leads to an improved effect of the augmentation. One issue of context extraction in the current work is, that it mostly takes into account the beginning of the sentence, which might cause problems when the most relevant information simply happens to be further back. An idea would be to first create a short summary of the sentence and use that as a cue for the generation model. In addition, we encourage using ContextGen on more datasets to further test effectiveness. Our results show that a pre-trained GPT2 model is capable of adapting even a technical language style given a context cue and we are confident that our approach will perform similarly on language domains different from the ones used in our experiments. On a final note, we designed ContextGen to be a cost effective baseline for model-based data augmentation. In this work, we employ a fairly large ensemble of 10 BERT classifiers, which is a fairly expensive approach in terms of computation. However, the size of the ensemble can be decreased. Primarily, the classifier ensemble is used for the label voting process (Section 3). Since we see in the results that depending on the dataset, label voting is not always necessary to achieve improvements we believe that using a smaller classifier ensemble is entirely justifiable and likely to produce similar results.

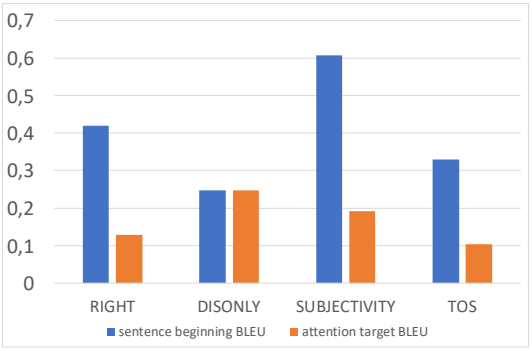

Figure 4: Average BLEU score of augmented sentences to original context sentences on each dataset. BLEU score is calculated for each augmented sentence and context pair and averaged over all sentences in the corpus.

6 CONCLUSION

In this paper we introduce ContextGen, a novel approach that utilizes data augmentation via generation without fine-tuning to address non-topical text classification tasks in low-resource settings. As we demonstrate in the experiments, our approach delivers improved prediction performance for non-topical classification when small training sets are available compared to the performance of the relevant baseline methods. The improvements are largest and especially beneficial for specialized domains, where large pre-trained models have limited success particularly for small training dataset coupled with non-topical classification tasks. We also analyze the generated sentences and show that generating content non-identical to the text used as context positively affects the overall classification performance. We can further assume that employing the classifiers self-attention to select words for the generations positively affects the augmentation performance. Finally, since our approach utilizes a large, pre-trained language model without any fine tuning it is also very efficient in comparison and can be employed when complex end-to-end architectures are difficult to train due to the limited availability of labelled data. We encourage the use of ContextGen for future experiments as an inexpensive baseline for classification tasks on specialized domains.

Acknowledgments
This research was funded and supported by IBM.

References

Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., Tepper, N., and Zwerdling, N. (2019). Not enough data? deep learning to the rescue!

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. cite arxiv:1810.04805Comment: 13 pages.

Fedus, W., Goodfellow, I., and Dai, A. M. (2018). Maskgan: Better text generation via filling in the blank.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. (2021). A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Jungiewicz, M. and Smywinski-Pohl, A. (2019). Towards textual data augmentation for neural networks: synonyms and maximum loss. *Computer Science*, 20(1).

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., and Socher, R. (2019). Ctrl: A conditional transformer language model for controllable generation.

Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Lippi, M., Pałka, P., Contissa, G., Lagioia, F., Micklitz, H.-W., Sartor, G., and Torroni, P. (2019). Claudette: an automated detector of potentially unfair clauses in online terms of service. *Artificial Intelligence and Law*, 27(2):117–139.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity. In *Proceedings of ACL*, pages 271–278.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (2002). Bleu: a method for automatic evaluation of machine translation.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rosario, B. and Hearst, M. A. (2004). Classifying semantic relations in bioscience texts. In *ACL*.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60.

Wei, J. and Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Wu, X., Lv, S., Zang, L., Han, J., and Hu, S. (2019). Conditional bert contextual augmentation. In Rodrigues, J. M. F., Cardoso, P. J. S., Monteiro, J., Lam, R., Krzhizhanovskaya, V. V., Lees, M. H., Dongarra, J. J., and Sloot, P. M., editors, *Computational Science – ICCS 2019*, pages 84–95, Cham. Springer International Publishing.

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., and Le, Q. V. (2018). Qanet: Combining local convolution with global self-attention for reading comprehension.

APPENDIX

code is available from: https://github.tik.uni-stuttgart.de/ac121207/ContextGen

Dataset	SUBJECTIVITY				
# sentences	50	100	200	500	1000
no augmentation	0.89	0.90	**0.92**	**0.93**	**0.94**
EDA	0.88	0.90	**0.92**	**0.93**	0.93
Lambada	0.80	0.89	0.91	0.91	**0.94**
ContextGen					
sentence-beginning context, no voting					
10% aug	0.89	**0.91**	0.91	**0.93**	**0.94**
50% aug	0.89	0.90	0.90	**0.93**	0.93
100% aug	0.88	0.90	0.90	0.92	0.93
sentence-beginning context, voting					
10% aug	0.87	0.90	0.91	0.94	**0.94**
50% aug	**0.90**	**0.91**	0.91	**0.93**	0.93
100% aug	0.89	0.90	0.90	0.92	0.93
attention target context, no voting					
10% aug	0.88	0.90	0.91	**0.93**	**0.94**
50% aug	0.88	0.88	0.90	**0.93**	**0.94**
100% aug	0.84	0.87	0.90	0.91	0.92
attention target context, voting					
10% aug	0.89	0.90	**0.92**	**0.93**	**0.94**
50% aug	0.88	**0.91**	0.92	**0.93**	0.93
100% aug	0.89	0.88	**0.92**	**0.93**	0.93

Table 4: ContextGen results on the SUBJECTIVITY dataset (common knowledge domain).

Model 1														
Dataset	RIGHT				DISONLY					TOS				
# sentences	50	100	200	314	50	100	200	500	1000	50	100	200	500	1000
sentence-beginning context														
augmentation %	100	50	50	50	10	10	10	50	100	50	10	10	50	50
voting	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	no	no	yes
attention-target context														
augmentation %	10	100	10	10	10	50	10	50	100	10	100	50	100	10
voting	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Model 2														
Dataset	RIGHT				DISONLY					TOS				
# sentences	50	100	200	314	50	100	200	500	1000	50	100	200	500	1000
sentence-beginning context														
augmentation %	100	10	50	10	10	10	10	50	100	100	10	50	10	10
voting	yes	yes	yes	yes	yes	yes	yes	no	no	yes	yes	yes	no	yes
attention-target context														
augmentation %	50	100	10	50	10	50	10	50	100	50	50	100	100	10
voting	yes	no	yes	yes	yes	yes	yes	no	yes	no	yes	yes	yes	yes
Model 3														
Dataset	RIGHT				DISONLY					TOS				
# sentences	50	100	200	314	50	100	200	500	1000	50	100	200	500	1000
sentence-beginning context														
augmentation %	100	100	50	10	10	10	10	100	100	100	10	50	50	50
voting	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	no	no	no
attention-target context														
augmentation %	50	100	10	10	50	50	10	100	100	10	50	50	100	10
voting	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes

Table 5: Configurations for best performing models from the dev set.

Dataset	RIGHT				DISONLY					TOS				
# sentences	50	100	200	314	50	100	200	500	1000	50	100	200	500	1000
best ContextGen (GPT-2)	**0.69**	**0.71**	**0.70**	**0.79**	**0.51**	**0.52**	0.54	**0.70**	**0.72**	**0.40**	**0.43**	**0.45**	**0.57**	**0.68**
best ContextGen (T5)	**0.69**	**0.72**	**0.72**	**0.79**	0.44	0.51	**0.57**	**0.70**	0.71	0.40	0.42	0.43	0.55	**0.68**

Table 6: Comparison of ContextGen using GPT-2 and T5 generation models.