GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation

Kang Min Yoo[1,2], **Dongju Park**[2], **Jaewook Kang**[2],
Sang-Woo Lee[1,2], and **Woomyeong Park**[2]

[1]NAVER AI Lab
[2]NAVER Clova AI
{kangmin.yoo, dongju.park, jaewook.kang}@navercorp.com
{sang.woo.lee, max.park}@navercorp.com

Abstract

Large-scale language models such as GPT-3 are excellent few-shot learners, allowing them to be controlled via natural text prompts. Recent studies report that prompt-based direct classification eliminates the need for fine-tuning but lacks data and inference scalability. This paper proposes a novel data augmentation technique that leverages large-scale language models to generate realistic text samples from a mixture of real samples. We also propose utilizing soft-labels predicted by the language models, effectively distilling knowledge from the large-scale language models and creating textual perturbations simultaneously. We perform data augmentation experiments on diverse classification tasks and show that our method hugely outperforms existing text augmentation methods. Ablation studies and a qualitative analysis provide more insights into our approach.

1 Introduction

In the seminal work by Brown et al. (2020), a large-scale language model, specifically GPT-3, has been shown to achieve superior performance on zero-shot and few-shot learning tasks by prompt-based in-context learning. In-context learning utilizes a prompt, which usually consists of a task description and few examples, to solve unseen tasks without the hefty price of fine-tuning. Recognizing the potential research applications of in-context learning and prompt-based control, a part of the NLP community has shifted its focus on understanding and devising advanced methods for optimizing prompt-based approaches (Schick and Schütze, 2020a; Shin et al., 2020; Zhao et al., 2021; Reynolds and McDonell, 2021).

However, these prompt-based approaches with inference on a large-scale language model suffer from several drawbacks. First, the number of in-context training examples is hard limited by the maximum prompt length enabled by the inherent language model architecture. Second, prompt-based approaches require online inference on the expensive large-scale language models. The inference may not be scalable in real-world use cases, because it is slow and incurs huge memory overhead. Lastly, the prompt-based approaches do away with conventional machine learning techniques, making it mostly incompatible with existing established fine-tuning methods.

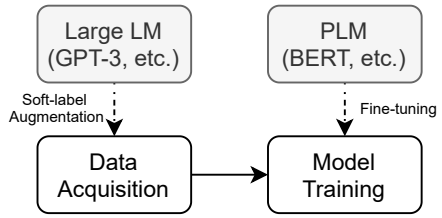

Figure 1: A conceptual diagram of text augmentation using large-scale language models.

To overcome such limitations, we propose a more practical solution to utilize large-scale language models for downstream NLP tasks. In our proposed framework, as depicted in Figure 1, large-scale language models are not used as the pre-trained model for further domain-adaptive fine-tuning nor the backbone for prompt-based in-context learning but for imbuing the original training set with synthetic text data.

We propose GPT3Mix, a method for generating synthetic but hyper-realistic text samples from a mixture of real samples utilizing large-scale language models such as GPT-3[1]. GPT3Mix extracts few sample sentences from the task-specific training data, embed these samples in the prompt, and generates an augmented mixed sentence influenced by the sample sentences. GPT3Mix uses soft-

[1]Despite what the name suggests, we can apply GPT3Mix to any large-scale autoregressive language models.

labels predicted by the large-scale language model to transfer knowledge of probability as in knowledge distillation (Hinton et al., 2015). In short, our method achieves both (1) *data augmentation* via generating synthetic examples inspired by existing data samples and (2) *knowledge distillation* by training smaller classification models using soft-labels predicted by the large language model.

Our approach takes inspiration from the mix-based data augmentation methods in the vision domain (Zhang et al., 2017). Several mix-based data augmentation methods are suggested for NLP models. One of the notable methods is MixText (Chen et al., 2020), in which BERT is used to generate novel augmentation samples from interpolated embedding spaces. However, despite its great success in the vision domain, deep-mixing text augmentation methods have seen limited effectiveness in real-world cases due to the difficulty of interpolating language from latent spaces (Bowman et al., 2016). Synthetic language interpolated from a model's hidden space such as the word embedding space of BERT may introduce noise, outweighing the benefit of novel sample discovery and causing deterioration in the training data distribution. Our work exploits the generative power of large-scale language models like GPT-3 to generate high-quality mixed samples from in-context examples.

We perform various data augmentation experiments on diverse classification tasks to verify our hypotheses and analyze our methodology. The contribution of our work is summarized as follows.[2]

1. We suggest employing prompt-based data augmentation using large-scale language models on top of the existing PLM fine-tuning paradigm to exploit the best of both worlds.

2. We propose GPT3Mix, a simple but effective text augmentation technique, that elicits knowledge and linguistic capability possessed by large-scale language models.

3. Our detailed analysis helps to understand the mechanism behind prompt-powered data augmentation, allowing us to better control the generation and the data augmentation behavior.

[2]The code to reproduce our results will be available at Github.

2 Related Work

Knowledge Distillation Knowledge distillation (Phuong and Lampert, 2019) is a technique that trains a smaller student classifier on the outputs of a larger teacher classifier. Knowledge distillation for language models in the context of model compression has been well-studied in the literature. There have been various distilled models and distillation methods proposed for pre-trained language models (Sanh et al., 2019; Tang et al., 2019). By utilizing soft-labels predicted by the large-scale language model, our approach helps to transfer knowledge to the downstream classifiers.

Text Augmentation Text augmentation refers to methods for perturbing the linguistic space without altering class labels to improve the robustness and generalizability of the downstream models. Data augmentation has been studied extensively in the NLP scene. Text augmentation in the current literature comes with two flavors: shallow and deep augmentation. The shallow data augmentation techniques inject locally plausible small noises into the linguistic space (words or phrases), in the hopes that the perturbations produce linguistically acceptable samples while maintaining label consistency. Two examples are EDA (Wei and Zou, 2019) and synonym replacement (Zhang et al., 2016).

Another class of augmentation techniques employs external language models to improve global coherence and consistency. The back-translation approach exploits semantic consistency in translation language pairs to generate novel paraphrases (Fadaee et al., 2017). In the more recent line of work, pre-trained language models, such as BERT (Devlin et al., 2019) or the sequence-to-sequence variant BART (Lewis et al., 2020), are used to obtain more diverse and linguistically correct augmentation samples. For example, BART has been proven to be effective in populating text samples for data-scarce labels (Kumar et al., 2020). Ng et al. (2020) proposed using masked language models as a denoising autoencoder to generate synthetic texts. Some other researchers have taken the direction of perturbing the latent spaces, optionally by introducing variational inference in the architecture (Xia et al., 2020b,a; Hou et al., 2018; Yoo et al., 2019).

On the other hand, inspired by the mix-up technique (Zhang et al., 2017) proposed for the vision domain, there have also been works to mix existing text samples to produce realistic augmen-

tation texts using statistical methods (Guo et al., 2020; Sun et al., 2020; Chen et al., 2020). Furthermore, pseudo-labeling, the act of annotating unlabeled data with model predictions (Lee et al., 2013; Reed et al., 2014), has been actively used in semi-supervised learning settings (Chen et al., 2020; Xie et al., 2020; Berthelot et al., 2019).

Large-scale Language Models Pre-trained transformer-based language models (Devlin et al., 2019; Lewis et al., 2020) have initiated a new paradigm in the NLP scene, changing the way we design NLP pipelines. With the recent development of mega-scale language models (Shoeybi et al., 2019; Brown et al., 2020), we are witnessing another shift in the paradigm, namely prompt-based NLP. These large language models are essentially few-shot learners, allowing them to be controlled through natural text. There has been a steep rise in the community's interest to better understand the prompt-based mechanisms (Reynolds and McDonell, 2021; Schick and Schütze, 2020a; Shin et al., 2020; Jiang et al., 2020; Zhao et al., 2021). Our work relies on the previous findings on prompt-based manipulation.

To the best of our knowledge, this work is the first to propose using the prompt-based approach to generate synthetic samples from large-scale language models for the purpose of text augmentation.

3 GPT3Mix

Mixup (Zhang et al., 2017) is a simple learning technique that has been shown to be effective in preventing memorization and improving generalizability for the vision domain. The technique has been very effective on image data, but it has been harder to establish a standard approach for texts due to the inherent sparse nature of linguistic distributions, which attributes to the challenges of identifying adversarial text examples (Li et al., 2017). Inspired by the technique, we propose GPT3Mix as a powerful yet simple method to generate highly fluent synthetic samples based on a data distribution.

The proposed method (Figure 2) consists of three steps: (1) selecting examples from the dataset, (2) constructing a GPT3Mix prompt from the selected examples and meta-information about the dataset, and finally (3) extracting augmentation from the language model generation. This section provides details about each step as follows.

Example Selection For simplicity, we confine the downstream task to text classification tasks. Given a classification task $\mathcal{T}$, the training dataset $\mathcal{D}$ is a set of text $\mathbf{x}$ and associated label y pairs: $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid 1 \leq i \leq N\}$.

We randomly choose k examples from $\mathcal{D}$ to be anchors. Large-scale language models are known to be highly sensitive to the choice and the order of examples in the prompt (Reynolds and McDonell, 2021; Zhao et al., 2021). We conjecture that by carefully choosing the examples, we are able to control the generated augmentation samples from the language model. We conduct qualitative analysis on the augmentation samples to confirm our hypothesis (§4.4.5).

In our implementation, we simply used uniform distribution to choose k examples: $p_s(i) = 1/N$. Otherwise stated, most experiments are carried out by setting $k = 2$ to simulate Mix-up. As found in our ablation studies (§4.4.1), $k = 2$ provides a good trade-off between cost and performance.

Prompt Construction Given a set of prompt examples $\mathcal{D}_e = \{(\mathbf{x}_i, y_i) \mid 1 \leq i \leq k\}$ sampled from $\mathcal{D}$, we formulate the prompt as follows.

A GPT3Mix prompt consists of a description header, an enumeration of text-label pairs of $\mathcal{D}_e$, and the augmentation prefix. An example of the prompt is shown in the appendix (§A). Our prompt has been designed carefully with the current literature findings of GPT-3 prompts (Reynolds and McDonell, 2021) in mind.

Specifically, the prompt follows the general template shown in the appendix, but has task-specific information to allow the large-scale language models to generalize better about the data distribution. These task indicators are unique to each task and provide meta-information of the task.

1. **Text Type** T: Meta-type of the input text $\mathbf{x}$. For example, in movie review sentiment analysis, the text type corresponds to `movie review`.

2. **Label Type** L: Meta type of the label class y. For the example above, the label type corresponds to `sentiment`.

3. **Label-token Verbalizer** $v : \mathcal{Y} \rightarrow \mathcal{V}$: Similar to the concept of verbalizers in the work of Schick and Schütze (2020b), the one-to-one mapping between the label classes $y \in \mathcal{Y}$ and

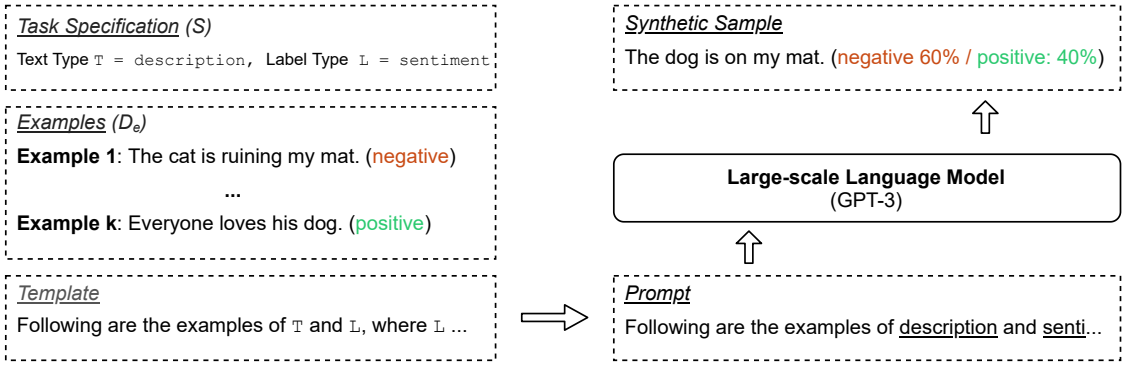

Figure 2: An illustration of GPT3Mix. The soft-labels of augmentation are extracted from the normalized label-token distributions predicted by the language model. Note that v has been omitted in the task specification S due to space limits.

word tokens in the language model's vocabulary $\mathcal{V}$ [3] is needed to formulate the prompt.

The triple of the meta information above forms the task specification $S = (T, L, v)$. Each task $\mathcal{T}$ requires a task specification $S_{\mathcal{T}}$ to be able to formulate a prompt for GPT3Mix. By default, the generic task specification $S_{\text{generic}} = (\text{text}, \text{label}, I)$ is used to construct prompts, where I is the identity function assuming that the class label exists in the vocabulary $\mathcal{V}$.

Augmentation Extraction As with other prompt-based approaches, the augmentation text $\mathbf{x}'$ and the label y' are generated in succession after the prompt as a natural text. A predefined prompt template in the examples signals the language model to generate $(\mathbf{x}', y')$ with a structure, allowing us to extracft respective values through pattern matching. Joint text and label generation also constraints the generated text to be associated with the correct label.

As illustrated in the prompt exhibit (§A), our particular prompt design ensures that the label token that corresponds to $v\left(y'\right)$ is generated after $\mathbf{x}$. This approach is inspired by the findings that, when inducing language models to come to a verdict, they require sufficient token lengths of "silent reasoning" prior to coming to a conclusion.

As large-scale language models are known to be few-shot learners (Brown et al., 2020), we also leverage GPT-3 to perform pseudo-labeling. The likelihood of generating the label-tokens is normalized to obtain the soft-label probability of the augmentation text $\mathbf{x}'$. Concretely, the pseudo-label probability of an augmentation text $\mathbf{x}'$ being labelled with label y' is as follows:

$$p\left(y' \mid \mathbf{x}'\right) \propto p_{\text{LM}}\left(v_{\mathcal{T}}\left(y'\right) \mid \mathcal{P}\left(\mathbf{x}', S_{\mathcal{T}}\right)\right), \quad (1)$$

where p_{LM} is the language modeling likelihood and $\mathcal{P} : \mathcal{S} \rightarrow \mathcal{X}$ is the function that constructs the prompt given a task specification.

Our approach effectively combines text perturbation, pseudo-labeling, and knowledge distillation in a single augmentation operation. In practice, augmentation samples with pseudo-labels are trained along with the real samples using the cross-entropy loss. This is in contrast to prior work, in which pseudo-labels are usually used for consistency regularization in the context of semi-supervised learning (Berthelot et al., 2019).

4 Experiments

We evaluate our augmentation approach on the following six classification benchmarks:

- **SST-2** (Socher et al., 2013): Stanford Sentiment Treebank is a sentiment classification dataset that contains movie reviews crawled from Rotten Tomatoes and their corresponding binary labels.
- **CR** (Hu and Liu, 2004): The Customer Review (CR) dataset is a set of Amazon product reviews and their binary labels.
- **COLA** (Warstadt et al., 2018): The Corpus of Linguistic Acceptability is a collection of sentences extracted from publications expertly annotated with grammaticality.

[3] In our implementation, we do not consider cases where a label class corresponds to multiple tokens. Regardless, expanding our work to incorporate multiple label tokens should be trivial.

		DistilBERT$_{\text{base}}$				BERT$_{\text{base}}$			
Dataset	**Sub-sample**	NoAug	EDA	BT	GPT3Mix	NoAug	EDA	BT	GPT3Mix
SST2	0.1%	$55.8_{5.1}$	$54.4_{4.4}$	$55.7_{5.2}$	$\mathbf{61.0}_{6.0}$	$56.52_{4.3}$	$56.9_{6.1}$	$58.0_{4.1}$	$\mathbf{63.3}_{4.0}$
	0.3%	$64.9_{8.0}$	$67.7_{7.9}$	$64.4_{7.5}$	$\mathbf{71.7}_{8.7}$	$70.7_{2.9}$	$70.0_{5.4}$	$64.8_{7.5}$	$\mathbf{78.0}_{5.6}$
	1.0%	$77.9_{3.6}$	$74.4_{5.0}$	$77.3_{2.8}$	$\mathbf{80.3}_{3.0}$	$\mathbf{82.5}_{1.8}$	$79.7_{1.1}$	$81.1_{3.5}$	$82.0_{2.7}$
COLA	0.1%	$64.9_{4.7}$	$59.0_{9.7}$	$51.4_{6.8}$	$\mathbf{68.5}_{0.2}$	$60.9_{6.7}$	$57.5_{5.0}$	$52.8_{4.2}$	$\mathbf{67.8}_{1.5}$
	0.3%	$62.2_{7.2}$	$66.1_{3.0}$	$57.3_{5.6}$	$\mathbf{68.5}_{0.2}$	$65.5_{3.0}$	$63.5_{5.6}$	$63.5_{4.8}$	$\mathbf{68.7}_{0.9}$
	1.0%	$67.8_{1.6}$	$64.0_{2.3}$	$61.8_{3.6}$	$\mathbf{69.2}_{0.5}$	$67.1_{4.8}$	$66.5_{5.8}$	$66.7_{2.7}$	$\mathbf{71.0}_{1.8}$
TREC6	0.1%	$29.3_{5.0}$	$29.0_{4.6}$	$26.3_{5.4}$	$\mathbf{37.6}_{6.9}$	$\mathbf{35.7}_{8.2}$	$28.0_{9.4}$	$27.4_{5.1}$	$31.8_{5.4}$
	0.3%	$37.9_{5.9}$	$34.7_{7.1}$	$37.3_{8.0}$	$\mathbf{40.1}_{4.0}$	$37.9_{9.2}$	$35.8_{4.1}$	$\mathbf{38.8}_{8.7}$	$37.1_{5.4}$
	1.0%	$67.2_{4.9}$	$66.8_{4.9}$	$\mathbf{68.2}_{9.5}$	$65.8_{3.4}$	$67.6_{8.2}$	$69.6_{11.5}$	$\mathbf{72.6}_{8.0}$	$58.8_{2.9}$
CR	0.1%	$57.2_{5.7}$	$58.1_{9.8}$	$59.3_{3.3}$	$\mathbf{60.5}_{1.1}$	$56.9_{6.4}$	$56.3_{5.7}$	$56.9_{6.1}$	$\mathbf{61.7}_{1.5}$
	0.3%	$\mathbf{66.6}_{7.8}$	$63.0_{4.7}$	$60.9_{2.9}$	$65.5_{5.1}$	$65.7_{6.0}$	$64.4_{3.8}$	$63.7_{2.9}$	$\mathbf{66.9}_{3.0}$
	1.0%	$69.6_{6.3}$	$70.0_{3.0}$	$74.2_{6.9}$	$\mathbf{79.5}_{2.5}$	$77.1_{3.4}$	$74.4_{3.3}$	$76.2_{5.1}$	$\mathbf{80.7}_{1.6}$
SUBJ	0.1%	$\mathbf{83.7}_{5.1}$	$83.3_{2.9}$	$82.8_{5.4}$	$82.4_{2.8}$	$82.6_{2.8}$	$83.3_{5.4}$	$\mathbf{84.3}_{3.6}$	$83.8_{2.9}$
	0.3%	$88.5_{0.9}$	$\mathbf{89.2}_{0.6}$	$88.2_{1.0}$	$88.8_{1.7}$	$89.8_{0.8}$	$89.4_{1.3}$	$89.4_{2.0}$	$\mathbf{89.8}_{1.1}$
	1.0%	$90.7_{1.0}$	$90.7_{1.1}$	$90.5_{0.8}$	$\mathbf{91.4}_{0.4}$	$91.5_{1.1}$	$91.2_{1.2}$	$91.7_{0.5}$	$\mathbf{92.2}_{0.5}$
MPQA	0.1%	$\mathbf{69.7}_{1.3}$	$67.5_{3.7}$	$65.7_{7.0}$	$66.9_{2.5}$	$58.2_{4.9}$	$\mathbf{68.0}_{3.5}$	$59.4_{5.1}$	$61.7_{6.1}$
	0.3%	$76.6_{5.0}$	$\mathbf{78.0}_{2.3}$	$76.2_{3.9}$	$77.7_{5.8}$	$71.1_{2.6}$	$77.1_{3.1}$	$74.6_{4.2}$	$\mathbf{77.7}_{4.9}$
	1.0%	$83.7_{4.8}$	$82.3_{2.0}$	$81.9_{4.7}$	$\mathbf{85.3}_{3.2}$	$83.3_{2.1}$	$82.7_{2.1}$	$82.4_{3.2}$	$\mathbf{86.1}_{0.8}$
Average	0.1%	60.1	58.6	56.9	**62.8**	58.5	58.3	56.5	**61.7**
	0.3%	66.1	64.1	64.1	**68.7**	66.8	64.6	65.8	**69.7**
	1.0%	76.1	74.7	75.6	**78.6**	78.2	77.3	78.4	**78.5**

Table 1: Main data augmentation results on 0.1%, 0.3%, and 1.0% training set sub-sample levels. We compare different augmentation strategies by transformer architectures on the downstream classification performance. Experiments have been repeated 10 times and the statistics are presented in the mean$_{\text{std}}$ format.

- **TREC6** (Voorhees and Tice, 1999): The TREC dataset is a dataset for question classification consisting of open-domain, fact-based questions divided into broad semantic categories.

- **MPQA** (Wiebe et al., 2005): Multi-perspective Question Answering polarity datasets consists of opinions and their semantic polarity.

- **SUBJ** (Pang and Lee, 2004): Subjectivity dataset contains movie reviews annotated with binary labels of objectiveness.

4.1 Experimental Settings

To showcase our approach, we conduct downstream classification experiments on artificially data-scarce tasks by sub-sampling the training set. For each experiment, we perform a class-balanced sub-sample on the training set. We account for statistical variance in our experiments by fixating the sub-samples on 5 different data seeds and repeating the augmentation procedure and downstream classification experiments 10 times on the sub-samples. The data seeds were not screened and chosen by random[4].

For the classifier architecture, we use the base size BERT (Devlin et al., 2019) and DistilBERT (Sanh et al., 2019) models. For each downstream classification trial, we initialize the classifier model with the pre-trained parameters provided by the Huggingface Transformers library (Wolf et al., 2019) and randomly initialize the classifier layers, which consist of two fully connected layers that predict the class labels from the output embeddings of the transformer architectures. The classifiers are trained automatically by employing early stopping against the validation score with patience of 20 training epochs.

4.2 Implementation Details

For selecting the optimal task specification for each task in GPT3Mix augmentation, we evaluated the performance of few handcrafted task specification candidates on the validation set and chose the highest performing one. The details about the optimal task specifications are presented in §B. The inference on GPT-3 was carried out via the OpenAI API Beta Access program. We used the largest GPT-3 model available on (davinci) unless other-

[4]The data seeds were generated using a master seed.

	k			
Sub.	1	2	4	8
0.1%	$65.5_{3.3}$	$71.2_{6.5}$	$74.6_{3.9}$	$72.0_{6.7}$
0.3%	$78.9_{3.9}$	$80.0_{2.7}$	$80.2_{2.1}$	$80.0_{1.6}$
1.0%	$85.2_{0.6}$	$84.3_{0.7}$	$84.3_{0.7}$	$84.2_{1.2}$

Table 2: An ablation study on the number of examples k in GPT3Mix prompts. When $k = 1$, GPT-3 produces point-wise perturbed samples. Experiments are carried out on the SST-2 dataset.

	Model Size			
Sub.	ada	babbage	curie	davinci
0.1%	$61.9_{4.1}$	$65.2_{6.9}$	$65.9_{5.3}$	$67.6_{7.2}$
0.3%	$74.6_{4.8}$	$69.7_{7.3}$	$74.6_{4.5}$	$78.3_{2.9}$
1.0%	$81.6_{1.0}$	$82.5_{1.1}$	$83.4_{1.8}$	$84.3_{1.1}$

Table 3: An ablation study on the size of the language model with the SST-2 dataset. Larger language models provide greater augmentation benefits in data-limited environment.

wise stated. On average, a GPT3Mix augmentation roughly consumes 300 tokens in combined length (prompt and generation).

The augmentation ratio between the training set and the synthetic set was set to 1 unless otherwise stated.

During classifier training, we used the Adam optimizer with decoupled weight decay (Kingma and Ba, 2014; Loshchilov and Hutter, 2017) and a learning rate of 3e-5. The learning rate had a warm-up period of 3 epochs. PyTorch and M40 GPUs were used to run the experiments.

4.3 Data Augmentation Experiments

We compare our approach to Easy Data Augmentation (EDA) (Wei and Zou, 2019) and back-translation (BT) (Fadaee et al., 2017). For the back-translation baseline, texts were translated to and from German using Transformer architectures trained on the WMT16 English-German corpus provided by Fairseq (Ott et al., 2019).

The results on data-scarce text augmentation are presented in Table 1. First, we notice that, in most cases, our approach outperforms other augmentation baselines by a large margin. Also, our approach offers much more performance stability in terms of the variance of repeated trials and inter-task fluctuations than any other augmentation methods. Although back-translation and EDA do outperform GPT3Mix at certain configurations (e.g. CR 0.3%, TREC6 1.0%), GPT3Mix offers the most stable and consistent performance boost for the downstream classifiers across all tasks. This is more evident when we observe the average downstream classification accuracies of all tasks, in which GPT3Mix improves the classification accuracies by 2-3% on average, whereas EDA and BT perform worse than the baseline without augmentation[5].

Note that the results for GPT3Mix can be improved further by increasing the augmentation ratio beyond 1 (refer to §4.4). As our API budget increases over time, we anticipate achieving better results as well in the future.

Full Dataset Experiments We also perform full dataset data augmentation experiments to confirm that GPT3Mix still offers benefits even when task-specific data are abundant. We augmented the full SST-2 dataset with 1:1 ratio of synthetic samples from GPT3Mix, and our experiments on the downstream classification show that GPT3Mix improves the accuracy of $\text{DistilBERT}_{\text{base}}$ from 90.28% to 90.70% (0.42% improvement) and the accuracy of $\text{BERT}_{\text{base}}$ from 90.33% to 93.25% (2.92% improvement). A larger improvement margin is observed in the architecture with the greater expressive power, and this observation is in agreement with prior work (Zhang et al., 2017; Shafahi et al., 2019).

4.4 Ablation Studies

We conduct a number of ablation experiments to study the underlying mechanism of GPT3Mix. Note that the augmentation results for GPT3mix in the following ablation studies outperform the results presented in §4.3 due to ablation studies having higher augmentation ratios[6].

4.4.1 Number of Prompt Examples

First, the effect of the number of examples in GPT3Mix prompts (k) on the downstream augmentation performance is studied. GPT3Mix requires

[5]We employed the well-established hyperparameters and the best hyperparameters proposed by the authors for EDA and BT. Augmentation ratios for all methods were set to 1 for a fair comparison with GPT3Mix.

[6]We limit the ratio in the main data augmentation experiments to the lowest level a due to the constrained API budget.

Example 1	Laughably, irredeemably awful. (*negative*)
Example 2	Well-made but mush-hearted. (*positive*)
GPT3Mix	Groundbreaking, disturbing. (***positive***: 75%, *negative*: 25%)
Example 1	Berry's saucy, full-bodied performance give this aging series a much needed kick, making "Die Another Day" one of the most entertaining Bonds in years. (*positive*)
Example 2	Moonlight Mile doesn't quite go the distance but the cast is impressive and they all give life to these broken characters who are trying to make their way through this tragedy. (*positive*)
GPT3Mix	"Gosford Park" takes a leisurely approach to story telling and along the way it mixes intrigue, comedy, and romance. (***positive***: 98%, *negative*: 2%)
Example 1	It's just not very smart. (*negative*)
Example 2	It's quite an achievement to set and shoot a movie at the Cannes Film Festival and yet fail to capture its visual appeal or its atmosphere. (*negative*)
GPT3Mix	Excessively talky, occasionally absurd and much too long, Munich is a fascinating mess. (*positive*: 21%, ***negative***: 79%)

Table 4: SST-2 augmentation samples from GPT3Mix (davinci).

$k \geq 2$ to effectively mix existing samples and generate interpolated text samples. However, supplying one example ($k = 1$) per prompt and expecting GPT-3 to introduce perturbations or paraphrases of the given example can be a viable strategy. We vary k on the SST-2 dataset and observe the downstream performances (Table 2). The second-largest GPT-3 model (curie) was used and the augmentation multiplier was set to 10.

From the results, we notice that when the data availability is severely limited (i.e. 0.1% and 0.3%), point-wise perturbation doesn't offer the performance improvement as much as when $k \geq 2$. However, as data becomes more abundant, increasing the number of mixing samples offers marginally small benefits for data augmentation. Yet, increasing the number of examples incurs additional overhead to the GPT-3 inference cost.

Generally, over-providing prompt examples may constraint the degrees of freedom and causing the synthetic samples to overfit on the data, hurting the downstream performances. However, a significant improvement from $k = 2$ to $k = 4$ is observed for the 0.1% sub-sample level. In our data augmentation studies, we weigh in on $k = 2$ as a reasonable balance between the trade-off between GPT-3 inference costs and performance gains.

4.4.2 Language Model Capacity

Next, we study the influence of the model capacity of the augmenting language model on the quality of augmentations. OpenAI offers GPT-3 in four different capacities: `ada`, `babbage`, `curie`, and `davinci`[7], listed in the increasing order of model complexity. In this study, the augmentation ratio is set to 5.

As expected, the results (Table 3) show that having larger and more expressive language models benefit data augmentation.

Sub.	NoAug	Hard Labels	soft-labels
0.1%	$55.8_{5.1}$	$61.6_{8.0}$	$71.2_{6.5}$
0.3%	$64.9_{8.0}$	$67.7_{5.9}$	$80.0_{2.7}$
1.0%	$77.9_{3.6}$	$79.0_{2.8}$	$84.3_{0.7}$

Table 5: An ablation study on the employment of pseudo-labels. Hard labels are obtained from the beam search of the entire sequence autoregressively generated by the language model.

4.4.3 Task Specification

We are also interested in how the design choice of task specification for prompt construction affects the downstream performance. To analyze the effect, we compare the optimal task specification $S_{\mathcal{T}^\star}$ to a generic one (S_{generic}). For this study, we used `curie` as the augmenting language model with an augmentation ratio of 3. The results in Table

[7]The sizes of the language models are known to be 2.7B, 6.7B, 13B, and 175B respectively; however, OpenAI has not officially disclosed the exact numbers yet.

Dataset	Sub.	NoAug	S_{generic}	$S_{\mathcal{T}^\star}$
SST2	0.1%	$55.8_{5.1}$	$60.1_{5.2}$	$71.2_{6.5}$
	0.3%	$64.9_{8.0}$	$72.6_{5.7}$	$80.0_{2.7}$
	1.0%	$77.9_{3.6}$	$81.4_{1.7}$	$84.3_{0.7}$
COLA	0.1%	$64.9_{4.7}$	$68.4_{0.4}$	$68.6_{0.0}$
	0.3%	$62.2_{7.2}$	$65.7_{2.7}$	$68.7_{0.2}$
	1.0%	$67.8_{1.6}$	$68.7_{0.3}$	$69.1_{1.1}$

Table 6: An ablation study on task specifications. S_{generic} denotes a generic task specification that does not hold task-specific meta-information (§3), and $S_{\mathcal{T}^\star}$ denotes the optimal specification for the corresponding task.

6 support our conjecture that the language model utilizes the meta-information about the dataset to generate better data samples, and thus prompt designs have a significant impact on the augmentation quality. However, the generic task specification outperforms other augmentation baselines, highlighting the effectiveness of employing large-scale language models as the augmentation source.

4.4.4 Pseudo-labeling

Finally, we study the effect of employing pseudo-labels from the label token probabilities predicted by the large-scale language model. we compare the augmentation performance when the label tokens optimized from the sequence-wide beam search are used instead. Results on SST-2 (Table 5) show that employing soft-labels has a strong advantage over sequence-optimized labels. The performance gap between the hard and soft-labels can be considered as the benefit of utilizing the class distribution jointly predicted by the language model as a form of knowledge distillation for synthetic samples (Kim and Rush, 2016). `curie` was used as the GPT-3 model with the augmentation ratio of 5.

4.4.5 Qualitative Analysis

Language models are known to be sensitive to the selection and the order of the examples presented in the prompt, causing the proceeding generation to exhibit bias (Zhao et al., 2021; Reynolds and McDonell, 2021). Our proposed method hinges on this unique property of large-scale language models, hence we intend to qualitatively examine the augmentation samples to support our hypothesis.

The augmentation samples for the SST-2 dataset are presented in Table 4. First, we notice that the sentiment of the synthetic sample strongly depends on the class labels of the examples. When both examples are either all *positive* or all *negative*, the sentiment of the augmentation sample is heavily influenced to have the corresponding bias. In the case of mixed class labels, the resulting class distribution is less peaky. Second, we also discover that the augmented sample follows the similar syntactic and semantic structure of the example texts. As demonstrated in the first case, the short and phrasal structure of the examples is well translated into the generated sample, supporting our claim that language models are able to learn from in-context examples, even for generation and pseudo-labeling tasks. Similar patterns are observed in the subsequent cases, in which even the entities and subjects of the examples are perturbed and carried over into the synthetic samples.

5 Conclusion

In this paper, we proposed a novel text augmentation technique called GPT3Mix that leverages large-scale language models and their abilities to perform controlled generation via prompts. Our extensive experiments on various classification tasks show that the augmentation samples can improve robustness and the classification performance of pre-trained transformer models, and the experiments suggest that GPT3-based data augmentation can become a competitive alternative to prompt-based task-solving (Brown et al., 2020) or direct fine-tuning (Liu et al., 2021). As future work, we are interested in achieving state-of-the-art results using existing pre-trained transformer architectures solely via GPT3Mix-enabled data augmentation. We are also working to improve data augmentation efficiency by example selection optimization and prompt construction optimization.

References

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157.

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.

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.

Demi Guo, Yoon Kim, and Alexander M Rush. 2020. Sequence-level mixed sample data augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5547–5552.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *stat*, 1050:9.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1234–1245.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2017. Robust training under linguistic adversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 21–27.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. Ssmba: Self-supervised manifold based data augmentation for improving out-of-domain robustness. *arXiv preprint arXiv:2009.10195*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278.

Mary Phuong and Christoph Lampert. 2019. Towards understanding knowledge distillation. In *International Conference on Machine Learning*, pages 5142–5151. PMLR.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. *arXiv preprint arXiv:2102.07350*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Timo Schick and Hinrich Schütze. 2020a. Exploiting cloze questions for few-shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Timo Schick and Hinrich Schütze. 2020b. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial training for free! *arXiv preprint arXiv:1904.12843*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, S Yu Philip, and Lifang He. 2020. Mixup-transformer: Dynamic data augmentation for nlp tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3436–3440.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

Ellen M Voorhees and Dawn M Tice. 1999. The trec-8 question answering track evaluation. In *TREC*, volume 1999, page 82. Citeseer.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

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.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Congying Xia, Caiming Xiong, Philip Yu, and Richard Socher. 2020a. Composed variational natural language generation for few-shot intents. *arXiv preprint arXiv:2009.10056*.

Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip Yu. 2020b. Cg-bert: Conditional text generation with bert for generalized few-shot intent detection. *arXiv preprint arXiv:2004.01881*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33.

Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. 2019. Data augmentation for spoken language understanding via joint variational generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7402–7409.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2016. Character-level convolutional networks for text classification.

Tony Z Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *arXiv preprint arXiv:2102.09690*.

Dataset	T	L	v
Generic	text	label	$\cdot \rightarrow \cdot$
SST-2	movie review	sentiment	`pos` →`positive`, `neg` →`negative`
CR	customer review	sentiment	`pos` →`positive`, `neg` →`negative`
SUBJ	text	objective	`subjective` →`no`, `objective` →`yes`
COLA	text	grammar	`acceptable` →`correct`, `unacceptable` →`incorrect`
TREC6	question	type	`ABBR` →`abbreviation`, `LOC` →`location`, `DESC` →`description`, `NUM` →`numeric` `ENTY` →`entity`, `HUM` →`human`
MPQA	text	sentiment	`pos` →`positive`, `neg` →`negative`

Table 7: Optimal task specifications.

A Prompts

The GPT3Mix prompt uses the following template. The template corresponds to the prompt-constructing function $\mathcal{P}$, which require a task specification $S_{\mathcal{T}} = (T, L, v)$.

```
Each item in the following list contains a <text type> and the
    respective <label type>. <label type> is one of '<label token 1>',
    ..., or '<label token N>'.

<text type>: <example text 1> (<label type>: <example label 1>)
...
<text type>: <example text k> (<label type>: <example label k>)
<text type>:
```

For example, given $S_{\text{SST2}} = (\text{movie review}, \text{sentiment}, I)$, the constructed GPT3Mix prompt is as follows.

```
Each item in the following list contains a movie review and the
respective sentiment. The sentiment is one of 'positive' or 'negative'.

Movie review: Despite its Hawaiian setting, the science-fiction
    trimmings and some moments of rowdy slapstick, the basic plot of
    ``Lilo'' could have been pulled from a tear-stained vintage Shirley
    Temple script. (Sentiment: Negative)
Movie review: And people make fun of me for liking Showgirls.
    (Sentiment: Negative)
Movie review:
```

B Task Specifications

After validating candidate task specifications for each task, we have selected the following for conducting our experiments (Table 7).

Providing incorrect or suboptimal specifications to the prompt may cause a large drop in augmentation qualities. For example, in the case of designing task specifications for the COLA dataset, when "linguistic acceptability" is used as the label type (instead of the optimal "grammar"), the downstream performance on the 0.1% sub-dataset drops to 38.8%, resulting in performance worse than the non-augmented baseline of 68.80%.