Generate, Annotate, and Learn: NLP with Synthetic Text

Xuanli He[1] **Islam Nassar**[1] **Jamie Kiros**[2] **Gholamreza Haffari**[1] **Mohammad Norouzi**[2]
[1]Monash University, Australia [2]Google Research, Brain Team, Canada
{xuanli.he1, gholamreza.haffari}@monash.edu, mnorouzi@google.com

Abstract

This paper studies the use of language models as a source of synthetic unlabeled text for NLP. We formulate a general framework called ''generate, annotate, and learn (GAL)'' to take advantage of synthetic text within knowledge distillation, self-training, and few-shot learning applications. To generate high-quality task-specific text, we either fine-tune LMs on inputs from the task of interest, or prompt large LMs with few examples. We use the best available classifier to annotate synthetic text with soft pseudo labels for knowledge distillation and self-training, and use LMs to obtain hard labels for few-shot learning. We train new supervised models on the combination of labeled and pseudo-labeled data, which results in significant gains across several applications. We investigate key components of GAL and present theoretical and empirical arguments against the use of class-conditional LMs to generate synthetic labeled text instead of unlabeled text. GAL achieves new state-of-the-art knowledge distillation results for 6-layer transformers on the GLUE leaderboard.

1 Introduction

There is an abundance of unlabeled data in the real world, but task-specific unlabeled data within the scope of a given machine learning problem can be challenging to find. For instance, one cannot easily find in-domain unlabeled text conforming to the input distribution of a specific Natural Language Processing (NLP) task from the GLUE benchmark (Wang et al., 2019c). Some NLP tasks require an input comprising a pair of sentences with a particular relationship between them. Moreover, classification datasets typically represent a tailored distribution of data and only include a limited number of class labels. If task-specific unlabeled data were available, one could adopt self-training (Yarowsky, 1995) to automatically annotate unlabeled data with pseudo labels to improve accuracy and robustness of classifiers (Xie et al., 2020; Carmon et al., 2019). In addition, one can use knowledge distillation (Hinton et al., 2015) on fresh task-specific unlabeled data to more effectively compress deep neural networks and ensembles (Buciluǎ et al., 2006; Chen et al., 2020a).

In the absence of task-specific unlabeled data, one could *retrieve* unlabeled examples from a large and diverse open-domain dataset (Du et al., 2020). However, such a retrieval-based approach may not scale to problems with complex input schemes, for example, sentence pairs with certain relations. Recent work (Yang et al., 2020; Kumar et al., 2020b) has considered the use of Language Models (LMs) like GPT-2 (Radford et al., 2019) as a means of data augmentation, showing the effectiveness of this approach for commonsense reasoning and classification tasks. Existing approaches often consider *class-conditional* generation, where the synthetic data is produced by con ditioning on a specified class label. However, it is unclear whether class-conditional generation is best suited for NLP tasks. Furthermore, existing pipelines often make synthetic data generation complicated as one needs to detect and discard low-quality synthetic *labeled* data or optionally re-label data (Yang et al., 2020; Vu et al., 2021b). For instance, Kumar et al. (2020b) observe that it is difficult for sentences generated by label-conditioned GPT-2 to retain the semantics/pragmatics of the conditioning label, leading to poor performance on downstream tasks.

We unify and simplify existing work on LMs as a data source for NLP and develop a general framework called ''generate, annotate, and learn (GAL)''. The generality of GAL allows us to use LM-generated synthetic data within novel applications such as Knowledge Distillation (KD) and few-shot learning. GAL builds on recent advances in text generation (Radford et al., 2019; Gao et al., 2021) and uses powerful LMs to synthesize task-specific unlabeled text by fine-tuning or conditioning a large LM on in-distribution examples. We use state-of-the-art classifiers to annotate generated text with soft pseudo labels when possible.

Transactions of the Association for Computational Linguistics, vol. 10, pp. 826–842, 2022. https://doi.org/10.1162/tacl_a_00492
Action Editor: André F.T. Martins. Submission batch: 2/2022; Revision batch: 4/2022; Published 8/2022.

We then combine labeled data and pseudo-labeled data to train more effective supervised models, resulting in significant gains on a range of NLP tasks like KD and few-shot learning.

We present a justification for GAL based on the empirical and vicinal risk minimization frameworks (Vapnik, 1992; Chapelle et al., 2001). We also investigate key components of GAL. We find that even if class-conditional LMs are available for text generation, it is more effective to discard the conditioning labels and let the teacher models produce pseudo labels. This observation is supported by our theoretical and empirical results. Accordingly, in contrast to prior work (Yang et al., 2020; Vu et al., 2021b), we advocate for the use of simple unconditional LMs for text synthesis. Further, we avoid any form of data filtering. Not surprisingly, we find that the diversity of synthetic text matters. That said, simple unconditional generation given random seeds provides sufficient diversity, and crafting diverse LM prompts is not needed.

In summary:

- We develop GAL, a simple and effective approach to the use of LMs for task-specific unlabeled text generation. We show that GAL can be used effectively for KD, self-training, and few-shot learning in NLP.
- We present theoretical and empirical investigations for GAL, explaining why it works and why using class-conditional LMs to generate synthetic labeled data is not as effective.
- GAL advances KD for NLP and establishes a new state-of-the-art (SoTA) resu lt for a single 6-layer transformer on the GLUE test set. It further improves prompt-based few-shot learning, providing an average improvement of 1.3% on four 4-shot learning NLP tasks, outperforming GPT-3-6B.

2 Related Work

Data synthesis with large pre-trained language models is closely related to our work (Kumar et al., 2020b; Yang et al., 2020; Vu et al., 2021b; Norouzi et al., 2020). Yang et al. (2020) propose a complex scheme, including label-conditioned data generation, data relabeling, data filtering, and two-stage training, to utilize synthetic data. By contrast, we show that a simple mixture of the original data and synthetic unconditionally generated data can provide sizable gains. Furthermore, we show a broader use of generative models on KD and few-shot learning. Vu et al. (2021b) take a task augmentation approach and employ conditional generation to produce in-domain synthetic data for an auxiliary language inference (NLI) task, which is then used to initialize the target-task classifier. However, not all tasks (e.g., grammatical acceptability judgments) can benefit from the NLI-style auxiliary task (Wang et al., 2019a). We aim to directly generate the unlabeled in-domain data for the target task. Unlike Norouzi et al. (2020), we do not use instance-based generative models.

More broadly, there has been a recent surge in data synthesis and augmentation in NLP, including rule-based and model-based approaches; see Feng et al. (2021) for a recent survey. Data synthesis with grammars has been explored in semantic parsing and natural language understanding (e.g., see Wang et al., 2015, 2021; Marzoev et al., 2020). Existing approaches to data augmentation for NLP include lexicon replacement, sentence retrieval, and round-trip machine translation (Wang and Yang, 2015; Yu et al., 2018; Kobayashi, 2018; Wu et al., 2019; Lichtarge et al., 2019; Wei and Zou, 2019; Alberti et al., 2019; Du et al., 2020; Shen et al., 2020). We, instead, propose the use of unconditional autoregressive LMs for data augmentation. This is simple, flexible, and powerful.

Self-training is one of the oldest approaches for semi-supervised learning (Scudder, 1965, Fralick, 1967; Agrawala, 1970; Yarowsky, 1995; Eisner and Karakos, 2005; Ueffing et al., 2007; Du et al., 2020). Abney (2004) and Haffari and Sarkar (2007) have theoretically analyzed self-training for simple decision lists. Recent theoretical work analyzes self-training for linear models, often under the assumption that the data distribution is (nearly) Gaussian (Carmon et al., 2019; Raghunathan et al., 2020; Chen et al., 2020b; Kumar et al., 2020a; Oymak and Gulcu, 2020). Wei et al. (2021) prove that, under ''expansion'' and ''class separation'' assumptions, self-training can lead to more accurate neural network classifiers. We present a theoretical framing of GAL in terms of empirical and vicinal risk minimization (Vapnik, 1992; Chapelle et al., 2001).

Knowledge Distillation (KD) (Buciluǎ et al., 2006; Hinton et al., 2015) uses a procedure similar to self-training to distill knowledge of an expressive teacher model into a smaller student model. In contrast, self-distillation (Furlanello et al., 2018; Zhang et al., 2019; Mobahi et al., 2020) uses teacher and student models of equal size, hoping to iteratively refine class labels. Previous work uses unlabeled data (Buciluǎ et al., 2006) and adversarial training (Wang et al., 2018) to improve KD. We demonstrate that synthetic data generated by unconditional generative models can improve KD on NLP, outperforming strong KD baselines, which often add more complexity and additional hyperparameters (e.g., Sun et al., 2019a; Jiao et al., 2019; Xu et al., 2020, Rashid et al., 2021).

3 Generate, Annotate, and Learn (GAL)

Given a labeled dataset $L = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, we first train an unconditional domain-specific generative model $g(\boldsymbol{x})$ on $L_{\boldsymbol{x}} = \{\boldsymbol{x}_i\}_{i=1}^N$, and then use it to synthesize unlabeled data. Such synthetic unlabeled data is used within self-training and KD even in the absence of in-domain unlabeled data. We restrict our attention to basic KD and self-training methods, even though GAL can be combined with more sophisticated semi-supervised techniques, too.

The effectiveness of GAL depends on the fidelity and diversity of synthetic examples. If we had access to the oracle generative process, we would be able to obtain the best KD and SSL results, as if we had access to real task-specific unlabeled data. Our preliminary experiments suggest that large language models are particularly effective within the GAL framework. Hence, as shown in Figure 1, to build the best domain-specific language model, we adopt a large language model pretrained on lots of open-domain text, and fine-tune it on a given dataset's inputs, that is, $L_{\boldsymbol{x}}$, *ignoring class labels*. Both our theory and ablations confirm that ignoring class labels is a good idea (c.f., Section 4 and 5). Transferring the knowledge of large language models is particularly beneficial when a small input dataset $L_{\boldsymbol{x}}$ of text is available (Hernandez et al., 2021).

To improve computational efficiency of GAL, we do not generate unlabeled data on the fly, but generate as many unconditional samples as possible and store them in a synthetic unlabeled dataset U. We use soft pseudo labels within self-training and KD, as we empirically found it is more effective than using hard labels on synthetic data.

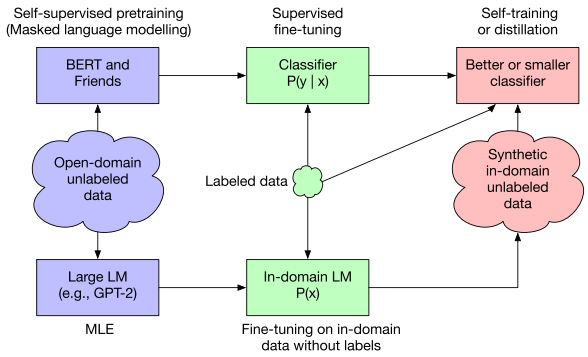

Figure 1: An illustration of GAL for NLP. We use open-domain data once for self-supervised pretraining (e.g., BERT) and once for training a large LM (e.g., GPT-2). BERT is fine-tuned on labeled data to yield a classifier for the task of interest. GPT-2 is fine-tuned on the same data without labels to obtain an unconditional task-specific LM, which is used to generate lots of synthetic in-domain unlabeled data for self-training and KD.

3.1 Knowledge Distillation with GAL

KD distills knowledge of an expressive teacher model into a smaller student model (Hinton et al., 2015). We pose the following objective function for KD with labeled and synthetic unlabeled data:

$$\ell_{kd} = \lambda\, \mathbb{E}_{(\boldsymbol{x},y)\sim L} H(y, f_s(\boldsymbol{x})) + (1-\lambda)\, \mathbb{E}_{\widetilde{\boldsymbol{x}}\sim g(\boldsymbol{x})} H(h(\widetilde{\boldsymbol{x}}), f_s(\widetilde{\boldsymbol{x}})), \quad (1)$$

where h is the *teacher* model, f_s is the *student* model, and g is the large pre-trained language model (e.g., GPT2) fine-tuned on the text in the training data $L_{\boldsymbol{x}}$. $H(q,p) = q^\top \log p$ is the softmax cross entropy loss. Note the use of $g(\boldsymbol{x})$, approximating the unknown real data distribution $P(\boldsymbol{x})$ in (1). Algorithm 1 summarizes the GAL-KD process.

3.2 Self-Training with GAL

Self-training encourages knowledge transfer between a *teacher* and a *student* model in such a way that the student can outperform the teacher. Algorithm 2 summarizes the GAL -self-training process. Given the labeled dataset L and the synthetic unlabeled dataset U, an initial model denoted f_1 is trained using supervised learning on the labeled dataset L. Then, at iteration t, one adopts

Algorithm 1 GAL-KD(L, g_0, f_0, h, k)

Input: Labeled dataset $L = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$
Initial parameters of a generative model g_0
Initial parameters of a classifier f_0
A teacher model h
Output: A well-trained student classifier f_s after KD
▷ unlabeled data generation
train a generative model g by fine-tuning g_0 on $L_{\boldsymbol{x}}$ where $L_{\boldsymbol{x}} = \{\boldsymbol{x} \mid (\boldsymbol{x}, y) \in L\}$
generate $U = \{\widetilde{\boldsymbol{x}}_j\}_{j=1}^{kN}$ by drawing kN random samples *i.i.d.* from $g(\boldsymbol{x})$
▷ knowledge distillation
apply h to unlabeled instances of U to get U'
train f_s by fine-tuning f_0 on $L \cup U'$
return f_s

Algorithm 2 GAL-self-training(L, g_0, f_0, k, T)

Input: Labeled dataset $L = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$
Initial parameters of a generative model g_0
Initial parameters of a classifier f_0
Output: A better self-training classifier f_{T+1} after T steps
▷ unlabeled data generation
train a generative model g by fine-tuning g_0 on $L_{\boldsymbol{x}}$ where $L_{\boldsymbol{x}} = \{\boldsymbol{x} \mid (\boldsymbol{x}, y) \in L\}$
generate $U = \{\widetilde{\boldsymbol{x}}_j\}_{j=1}^{kN}$ by drawing kN random samples *i.i.d.* from $g(\boldsymbol{x})$
▷ self-training
train a base model f_1 by fine-tuning f_0 on L
for $t = 1$ to T do:
  apply f_t to unlabeled instances of U to get U'
  train f_{t+1} by fine-tuning f_0 on $L \cup U'$
return f_{T+1}

f_t as the teacher model to annotate the unlabeled dataset U using *pseudo labels*. In self-training GAL, the student model f_{t+1} is trained to optimize a classification loss on the combination of L and U:

$$\ell_{t+1} = \lambda\, \mathbb{E}_{(\boldsymbol{x},y)\sim L} H(y, f_{t+1}(\boldsymbol{x})) + (1-\lambda)\, \mathbb{E}_{\widetilde{\boldsymbol{x}}\sim g(\boldsymbol{x})} H(f_t(\widetilde{\boldsymbol{x}}), f_{t+1}(\widetilde{\boldsymbol{x}}))\,, \quad (2)$$

where $\lambda = 0.5$ unless stated otherwise. Although many different variants of the basic self-training algorithm discussed above exist in the literature, we adopt the simplest variant of self-training and limit hyperparameter tuning to a bare minimum.

3.3 Domain-Specific Text Generation

We take a pretrained GPT-2 language model (Radford et al., 2019) and fine-tune it separately on each dataset of interest after removing class labels. We find that training from scratch on these datasets is hopeless, but the larger the pretrained GPT-2 variant, the better the validation perplexity scores are. For tasks modeling a relationship between multiple sentences, we concatenate a separator token [SEP] between consecutive sentences. To alleviate an over-fitting on the training set, we use the best checkpoint evaluated on the dev set as our generation engine. Once a fine-tuned GPT-2 model is obtained, we generate new domain-specific data by using top-*k* random sampling similar to Radford et al. (2019). We do not feed any prompt to the LM, but a special [BOS] token to initiate the generation chain. A generation episode is terminated when a special [EOS] token is produced. We generate diverse sentences by varying the random seed. After collecting enough synthetic data, we only retain unique sentences. For tasks with α input sentences, we discard generated samples that violate this constraint (approximately 10% of samples were rejected). Finally, we obtain task-specific synthetic data up to $40\times$ larger than the original training sets. For some samples of generated text for GLUE see Tables 11 and 12. We believe using bigger LMs and larger synthetic datasets will improve our results, but we are constrained by computer resources.

4 An Empirical Risk Minimization Perspective

In supervised learning, one seeks to learn a mapping f that, given an input $\boldsymbol{x}$, predicts a reasonable output y. To define the supervised learning problem formally, one assumes that input-output pairs are drawn from a joint distribution P, namely, $(\boldsymbol{x}, y) \sim P(\boldsymbol{x}, y)$, and a loss function $H(y, f(\boldsymbol{x}))$ is used to assess the quality of a mapping f. This loss is used to define a notion of *expected risk*:

$$R(f) = \mathbb{E}_{P(\boldsymbol{x},y)} H(y, f(\boldsymbol{x}))\,. \quad (3)$$

In almost all practical applications $P(\boldsymbol{x}, y)$ is unknown. Hence, a labeled dataset of examples $L = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ is used to approximate $R(f)$ as

$$\widehat{R}(f) = \frac{1}{N} \sum\nolimits_{i=1}^{N} H(y_i, f(\boldsymbol{x}_i))\,. \quad (4)$$

This objective function is known as *empirical risk*, and learning f through minimizing $\widehat{R}(f)$ is

known as the *empirical risk minimization* principle (Vapnik, 1992). To compensate for the finite sample size in (4), one typically combines $\widehat{R}(f)$ with a regularizer to improve generalization.

Beyond Empirical Risk Minimization. Empirical risk minimization (4) is motivated as a way to approximate $P(\boldsymbol{x}, y)$ through a set of Dirac delta functions on labeled examples: $P_\delta(\boldsymbol{x}, y) = \sum_i \delta(\boldsymbol{x} = \boldsymbol{x}_i, y = y_i)/N$. However, this approximation is far from perfect, hence one uses a heldout validation set for early stopping and hyperparameter tuning.

Vicinal risk minimization (Chapelle et al., 2001) approximates expected risk as $\mathbb{E}_{P_\nu(\boldsymbol{x},y)} H(y, f(\boldsymbol{x}))$, using a *vicinity distribution*, for example, $\nu(\tilde{\boldsymbol{x}}, \tilde{y} \mid \boldsymbol{x}, y) = \mathcal{N}(\tilde{\boldsymbol{x}} - \boldsymbol{x}, \sigma^2)\delta(\tilde{y} = y)$ to approximate $P(\boldsymbol{x}, y)$ as

$$P_\nu(\boldsymbol{x}, y) = \frac{1}{N} \sum_{i=1}^{N} \nu(\tilde{\boldsymbol{x}} = \boldsymbol{x}, \tilde{y} = y \mid \boldsymbol{x}_i, y_i) \; . \tag{5}$$

The goal is to increase the support of each labeled data point and improve the quality and robustness of the risk function.

Recent work on mixup regularization (Zhang et al., 2018) proposes an effective way to construct another vicinity distribution by interpolating between two data points and their labels. Despite their simplicity, these smoothing techniques tend to improve matters.

Generative Models for Risk Minimization. One can factorize the joint distribution of input-output pairs as $P(\boldsymbol{x}, y) = P(\boldsymbol{x})P(y \mid \boldsymbol{x})$. Accordingly, if one is able to learn a reasonable unconditional generative model of $\boldsymbol{x}$ denoted $g(\boldsymbol{x})$, then one can draw a pair $(\boldsymbol{x}, y)$ by first drawing $\boldsymbol{x} \sim g(\boldsymbol{x})$ and then using the current instance of f_t to draw $y \sim f_t(\boldsymbol{x})$. Then, one can use f_t and g to approximate expected risk as

$$R_t(f_{t+1}) = \mathbb{E}_{\boldsymbol{x} \sim g(\boldsymbol{x})} \mathbb{E}_{y \sim f_t(\boldsymbol{x})} H(y, f_{t+1}(\boldsymbol{x})) \; . \tag{6}$$

The quality of this approximation highly depends on the quality of f_t and g. If f_t is far from an optimal classifier f^* or $g(\boldsymbol{x})$ is far from $P(\boldsymbol{x})$, (6) yields a poor approximation.

The expected risk in (6) smoothens the risk landscape in complex ways beyond simple Gaussian smoothing and interpolation. This smoothing is applicable to any continuous, discrete, or structured domain as long as expressive generative models of $P(\boldsymbol{x})$ are available. That said, for almost all reasonable loss functions H (e.g., softmax cross entropy and squared error), (6) is minimized when $f_{t+1} = f_t$, which is not ideal, especially when f_t is far from f^*. On the other hand, empirical risk (4) anchors the problem in real labeled examples that are provided as ground truth.

GAL -self-training aims to combine the benefits of (4) and (6) via:

$$R_t(f_{t+1}) = \frac{\lambda}{N} \sum_{i=1}^{N} H(y_i, f_{t+1}(\boldsymbol{x}_i)) + (1-\lambda)\mathbb{E}_{\boldsymbol{x} \sim g(\boldsymbol{x})} \mathbb{E}_{y \sim f_t(\boldsymbol{x})} H(y, f_{t+1}(\boldsymbol{x})). \tag{7}$$

In this formulation, if f_t represents the minimizer of empirical risk (4), then $f_{t+1} = f_t$ is the minimizer of (7), too. However, one does not seek the global minimizer of empirical risk, but rather the best performance on heldout data. If f_t is obtained by stochastic gradient descent on any risk function, but early-stopped according to empirical risk on a heldout set, then using such f_t in (7) to define $R_t(f_{t+1})$ promotes the selection of a mapping f_{t+1} that minimizes empirical risk while staying close to the best performing mapping so far (i.e., f_t). This formulation motivates self-training and GAL as regularizers in the functional space and explains why they can conceivably work. Although the arguments are provided here for GAL-self-training, extending them to GAL-KD is straightforward (omitted due to the space constraints).

How About Class-conditional Generative Models? One can also factorize the joint distribution $P(\boldsymbol{x}, y)$ as $P(y)P(\boldsymbol{x} \mid y)$ and accordingly utilize a class-conditional generative model $g(\boldsymbol{x} \mid y)$ to derive the following expected risk formulation:

$$R(f) = \mathbb{E}_{y \sim P(y)} \mathbb{E}_{\boldsymbol{x} \sim g(\boldsymbol{x}|y)} H(y, f_{t+1}(\boldsymbol{x})) \; . \tag{8}$$

In this setting pseudo labeling is not needed as synthetic data is already labeled. One can show that the optimal classifier f_g^* that minimizes (8) for the cross-entropy loss is given by

$$f_g^*(y \mid \boldsymbol{x}) = g(\boldsymbol{x}|y)P(y) \Big/ \sum_{y'} g(\boldsymbol{x}|y')P(y') \; , \tag{9}$$

that is, turning the class-conditional generative model into a classifier by using the Bayes rule yields the optimal solution.

Model	MNLI(m/mm)	CoLA	SST-2	MRPC	STS-B	QQP	QNLI	RTE	Avg
Previous work:									
BERT-Theseus	82.4/82.1	47.8	92.2	87.6/83.2	85.6/84.1	71.6/89.3	89.6	66.2	78.6
BERT-PKD	81.5/81.0	–	92.0	85.0/79.9	–	70.7/88.9	89.0	65.5	–
tinyBERT	84.6/83.2	51.1	93.1	87.3/82.6	85.0/83.7	71.6/89.1	90.4	70.0	79.8
MATE-KD	86.2/85.6	58.6	95.1	91.2/88.1	88.5/88.4	73.0/89.7	92.4	76.6	83.5
Our results:									
DistilRoBERTa	83.8/83.4	55.9	93.2	87.4/83.1	87.5/87.5	71.7/89.1	90.6	73.3	81.2
DistilRoBERTa + KD	84.5/84.1	53.0	93.5	88.9/85.1	88.0/87.4	71.9/89.2	91.0	75.0	81.5
DistilRoBERTa + WS	86.2/85.9	52.2	94.0	89.9/86.4	88.7/88.3	71.7/89.2	91.5	76.2	82.1
DistilRoBERTa + RT	86.2/85.6	55.0	94.9	90.1/86.5	89.2/88.9	72.5/89.7	92.1	77.2	82.9
DistilRoBERTa + GAL	86.9/86.4	58.6	95.3	91.6/88.7	89.9/89.5	73.0/89.9	92.7	79.7	84.3

Table 1: GLUE test results for a 6-layer transformer. GAL establishes a new state of the art on KD for NLP. Baselines: BERT-Theseus (Xu et al., 2020), BERT-PKD (Sun et al., 2019a), tinyBERT (Jiao et al., 2019), MATE-KD (Rashid et al., 2021), DistilRoBERTa (Sanh et al., 2019), and DistilRoBERTa + KD (standard KD), DistilRoBERTa + WS (word substitution), and DistilRoBERTa + RT (round-trip translation). MNLI-m and MNLI-mm indicate matched and mismatched, respectively.

Provided that the accuracy of generative classifiers on text classification is behind their discriminate counterparts (e.g., Ravuri and Vinyals, 2019), we think substituting (8) into (7) is not a good idea. Essentially, by substituting (8) into the classification objective, one is regularizing f to remain close to f_g^*, which is not an effective strategy if f_g^* is not competitive. This argument corroborates the evidence from our ablation studies and recent work showing that using class-conditional generative models to augment supervised learning does not provide big gains (Ravuri and Vinyals, 2019).

That said, one can still use class-conditional generative models to synthesize high-fidelity samples. As long as these samples are treated as unlabeled examples and annotated using a classifier, for example, f_t, we believe this is a reasonable approach falling under GAL. Note that our argument above only applies to the scenario that class-conditional generative models are used to synthesize labeled examples. In other words, GAL emphasizes prediction of the labels in the course of the algorithm, rather than having the labels predefined. If one uses the unlabeled synthetic examples from class-conditional generative models, it still aligns to (7), which will be verified in Section 5.4.

5 Experiments

In this section, we assess the effectiveness of GAL on KD, self-training, and few-shot learning.

5.1 State-of-the-art Results of Knowledge Distillation with GAL on GLUE

We use the GLUE benchmark (Wang et al., 2019c) for our KD experiments; see Appendix A.1 for benchmark details. Our synthetic unlabeled dataset U includes 40× as many examples as the original dataset for each task in GLUE.

It is known that KD on fresh data, unseen during training, performs better (Buciluǎ et al., 2006; Chen et al., 2020a) than KD on original training data. Hence, we investigate the effectiveness of KD using generated unlabeled data through GAL.

We use the HuggingFace implementation (Wolf et al., 2020) for KD experiments and adopt a standard experimental setup consistent with previous work (Sun et al., 2019a; Xu et al., 2020). Following Rashid et al. (2021), fine-tuned RoBERTa-large (24-layer transformer) represents the teacher and a DistilRoBERTa (6-layer transformer) (Sanh et al., 2019) is used as the student. We train the student model on U and L, where U is annotated by the best RoBERTa-large model, achieving an average score of 86.5. We then mix L and U at a ratio of 1:4, which is equivalent to $\lambda = 0.2$. This ratio works best on the dev set.

Table 1 shows the results of individual 6-layer transformers on the GLUE test set. All of the baselines use an identical student architecture. GAL achieves the best entry on the GLUE leaderboard, marking a new state-of-the-art for KD on NLP. It outperforms strong KD baselines such as DistilRoBERTa (Sanh et al., 2019), BERT-PKD (Sun

Model	MNLI	CoLA	SST-2	MRPC	STS-B	QQP	QNLI	RTE	Avg
RoBERTa base	$87.7_{0.1}$	$63.6_{0.4}$	$94.8_{0.1}$	$90.1_{0.4}$	$90.8_{0.1}$	$91.5_{0.1}$	$92.6_{0.1}$	$78.8_{0.4}$	86.2
+GAL (iter 1)	$87.9_{0.1}$	$65.1_{0.5}$	$95.3_{0.1}$	$91.7_{0.5}$	$91.4_{0.1}$	$91.8_{0.1}$	$93.1_{0.1}$	$81.4_{0.4}$	87.2
+GAL (iter 2)	$88.0_{0.1}$	$65.2_{0.5}$	$95.3_{0.1}$	$92.2_{0.4}$	$91.5_{0.1}$	$91.7_{0.1}$	$93.2_{0.1}$	$82.4_{0.5}$	**87.4**
+GAL (iter 3)	$87.9_{0.1}$	$65.5_{0.5}$	$95.3_{0.1}$	$92.2_{0.5}$	$91.7_{0.2}$	$91.7_{0.1}$	$93.2_{0.1}$	$82.0_{0.5}$	**87.4**
RoBERTa base + self-distillation	$88.1_{0.1}$	$63.7_{0.5}$	$95.2_{0.1}$	$90.3_{0.4}$	$90.4_{0.1}$	$91.5_{0.1}$	$93.1_{0.1}$	$79.7_{0.5}$	86.5

Table 2: RoBERTa base and GAL self-training results on GLUE dev sets, averaged across 5 independent runs (numbers in the subscript indicate the error bar, i.e., standard deviation divided by $\sqrt{5}$.).

et al., 2019a), BERT-Theseus (Xu et al., 2020), tinyBERT (Jiao et al., 2019), and MATE-KD (Rashid et al., 2021). It also outperforms our own DistilRoBERTa+KD baseline, which learns from soft labels produced by an identical RoBERTa-large ensemble on the original labeled dataset. While the use of soft labels outperform the vanilla fine-tuned DistilRoBERTa model, it significantly underperforms our KD+GAL baseline. We also compare with two strong data-augmentation baselines, round-trip translation (RT) (Yu et al., 2018; Shleifer, 2019) and word substitutions (WS) (Jiao et al., 2019; Wei and Zou, 2019). For RT, We generate 40× unlabeled data using German as the bridge language (English→German→English). The translations are generated via the best model in WMT19 (Ng et al., 2019). We use the codebase from Jiao et al. (2019) to conduct WS data augmentation. We mirror the KD experimental setup of GAL for both RT and WS. Although DistilRoBERTa+RT and DistilRoBERTa+WS are better than vanilla DistilRoBERTa and KD variants, they still drastically underperform our approach.

5.2 Self-Training with GAL on GLUE

We fine-tune a pretrained RoBERTa model provided by fairseq (Ott et al., 2019) on each GLUE task. Fine-tuned RoBERTa serves as the first teacher model for self-training. Each student model is initialized with the original pretrained RoBERTa and fine-tuned with exactly the same hyperparameters as suggested by fairseq (Ott et al., 2019). We combine the labeled dataset L and the synthetic dataset U with a ratio of 1:1, by oversampling labeled data. This corresponds to $\lambda = 0.5$ in Eq. (7).

Table 2 shows that GAL provides an average improvement of +1.3% over RoBERTa-base. We see consistent improvements with more GAL iterations, but performance saturates after three iterations. We further compare our approach with a self-distillation (Furlanello et al., 2018) baseline, in which the teacher and student models use the same architecture and transfer knowledge via the original labeled training set. Although self-distillation provides a slight improvement, the gains from GAL are more significant.

We delve deeper and combine GAL self-training with RoBERTa-large and report test results for both single model and ensemble model in Table 3. We observe consistent gains coming from GAL on RoBERTa-large. Our results underperform the latest and largest LMs from the GLUE leaderboard, but we are optimistic that GAL can be effectively combined with enormous LMs to provide additional gains.

5.3 Prompt-based Few-shot Experiments

GPT3 (Brown et al., 2020) has introduced an optimization-free paradigm for few-shot learning for NLP. Without updating the parameters, large LMs can correctly predict the labels of the inputs by conditioning on a prompt, which consists of an instruction, a few labeled instances and a new unlabeled input. We apply GAL to prompt-based few-shot learning. Specifically, we present k labeled examples as a prompt to GPT-J (Wang and Komatsuzaki, 2021), an open-sourced re-implementation of GPT-3-6B, and generate m synthetic examples, followed by the corresponding labels. Note that to mitigate noisy outputs, the generation of each synthetic example only conditions on the original k labeled examples. Finally, we concatenate the original k examples and m synthetic examples, and conduct a $(k+m)$-shot learning experiment with GPT-J.

Brown et al. (2020) studied a total of 51 few-shot learning tasks. Studying all of these tasks is prohibitively expensive. Thus, we filter tasks by following these two steps. First, since generating m synthetic examples for each test instance is computationally expensive, we exclude tasks that

Model	MNLI(m/mm)	CoLA	SST-2	MRPC	STS-B	QQP	QNLI	RTE	Avg
Individual Models (our implementation):									
RoBERTa-large	90.1/89.7	63.8	96.1	91.2/88.3	90.9/90.7	72.5/89.6	94.5	85.9	86.5
RoBERTa-large + GAL	90.2/89.8	66.2	96.4	92.0/89.2	90.7/90.5	73.6/89.9	95.0	86.3	87.1
Ensemble Models (our implementation):									
RoBERTa-large	91.2/90.5	66.8	96.9	92.8/90.3	91.9/91.6	74.5/90.4	95.5	87.7	87.9
RoBERTa-large + GAL	91.0/90.7	67.9	97.1	93.1/90.8	91.6/91.4	74.5/90.4	95.8	88.2	88.2
State-of-the-art:									
RoBERTa-large	90.8/90.2	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.3	95.4	88.2	88.0
ELECTRA	91.3/90.8	71.7	97.1	93.1/90.7	92.9/92.5	75.6/90.8	95.8	89.8	89.2
T5	92.2/91.9	71.6	97.5	92.8/90.4	93.1/92.8	75.1/90.6	96.9	92.8	89.8
ERNIE	91.9/91.4	74.4	97.8	93.9/91.8	93.0/92.6	75.2/90.9	97.3	92.0	90.2
DeBERTa	91.9/91.6	71.5	97.5	94.0/92.0	92.9/92.6	76.2/90.8	99.2	93.2	90.3

Table 3: RoBERTa-large with GAL self-training and SoTA methods evaluated on GLUE test sets. The benefit of GAL on single models is larger than ensembles. It appears that self-training reduce the variance of models. Baselines including much larger models: RoBERTa-large (Liu et al., 2019), ELECTRA (Clark et al., 2020), T5 (Raffel et al., 2020), ERNIE (Sun et al., 2019b), and DeBERTa (He et al., 2020). MNLI-m and MNLI-mm indicate matched and mismatched, respectively.

Model	SST-2	PIQA	COPA	BoolQ	Avg
4-shot	89.8 $_{0.8}$	76.0 $_{1.4}$	79.0 $_{1.5}$	64.3 $_{0.8}$	77.3
8-shot	91.3 $_{0.8}$	76.2 $_{1.2}$	79.0 $_{1.5}$	66.2 $_{0.8}$	78.2
16-shot	92.7 $_{0.6}$	77.0 $_{0.9}$	81.0 $_{1.1}$	66.8 $_{0.8}$	79.4
4-shot + synthetic 12-shot (GAL)	91.5 $_{0.7}$	76.7 $_{1.0}$	80.0 $_{1.2}$	65.9 $_{0.8}$	78.5

Table 4: Few-shot learning results for GPT-J (6B) (Wang and Komatsuzaki, 2021) on four NLP datasets. Accuracy is reported for these datasets.

have more than 5k test examples. Second, we filter tasks on which GPT-3-6B achieves a score lower than 65% (please refer to Table H.1 in Brown et al. [2020] for more details). After applying the filtering steps, we use four datasets: SST-2 (Wang et al., 2019c), PIQA (Bisk et al., 2020), COPA, and BoolQ (Wang et al., 2019b) as the testbed. We notice that in order to generate valid synthetic data, GPT-J requires to see at least 4 labeled examples. In addition, at most 16 examples of BoolQ can be fed into GPT-J without truncation. Thus, we set k and m to 4 and 12, respectively. As seen in Table 4, GAL leads to an average improvement of 1.2% over 4-shot learning, and reduces the gap between 4-shot and 16-shot learning. We noticed that the quality of some generated examples is low. We believe the performance of few-shot learning can be further improved with high-quality instances. One solution is to generate many synthetic examples, and select a high-quality subset. Since each test instance conditions on distinct labeled instances, one has to generate different synthetic instances for each test example from GPT-J, which causes expensive computation. Due to such computational constraints, we leave the investigation of data selection strategies to the future work.

5.4 Ablating Components of GAL on GLUE

We conduct an in-depth study of different components of GAL on GLUE datasets. Unless stated otherwise, we use a RoBERTa-base model with a combination of the original training data and $40\times$ synthetic data for each self-training experiment.

GPT-2 Model Size. Radford et al. (2019) present a few variants of the GPT-2 model including *GPT-2*, *GPT-2-medium*, *GPT-2-large*, and *GPT-2-XL*. Larger GPT-2 models yield better perplexity scores and higher generation quality. We utilize these models except GPT-2-XL within the GAL framework to study the impact of the generative model's quality on downstream task's performance. Table 5 shows that regardless of the

GPT-2	SST-2	RTE	MRPC	CoLA
NA	94.8	78.8	90.1	63.6
small	**95.5**	81.3	90.9	63.9
medium	95.3	81.3	91.3	63.7
large	95.3	**81.4**	**91.7**	**65.1**

Table 5: GAL with various GPT-2 model sizes on GLUE dev sets. NA indicates a RoBERTa base model. We **bold** the best numbers.

Pseudo label	SST-2	RTE	MRPC	CoLA
hard	95.0	80.7	90.8	63.0
soft	**95.3**	**81.4**	**91.7**	**65.1**

Table 6: GAL with soft vs. hard pseudo labels on GLUE dev sets. We **bold** the best numbers.

GPT-2 model sizes, GAL consistently surpasses the vanilla RoBERTa base. Moreover, SST-2 and RTE datasets are not sensitive to the capacity of GPT-2, but higher quality synthetic text improves the results on MRPC and CoLA datasets. We leave investigation of GPT-2-XL and even larger LMs such as GPT-3 (Brown et al., 2020) to future work.

Soft vs. Hard Pseudo Label. We investigate the use of soft and hard pseudo labels within the GAL framework. The results in Table 6 suggest that GAL using soft pseudo labels is more effective than hard labels on the GLUE benchmark. This finding is compatible with the intuition that soft labels enable measuring the functional similarity of neural networks better (Hinton et al., 2015).

Class-conditional Synthetic Data Generation. Previous work (Kumar et al., 2020b; Ravuri and Vinyals, 2019) suggests that it is challenging to utilize labeled synthetic data from class-conditional generative models to boost the accuracy of text and image classifiers. Our theory in Section 4 points to the potential drawback of class-conditional synthetic data. We empirically study this phenomenon, by fine-tuning GPT-2 in a class-conditional manner. Then we utilize its synthetic examples in two different cases: 1) labeled synthetic examples and 2) unlabeled synthetic examples. Table 7 shows that not only do class-conditional LMs underperform unconditional LMs in our GAL framework, but also they are much worse than the baseline, when using the predefined labels. Nevertheless, if we apply GAL to these examples, the class-conditional LM is on par with the unconditional one, which corroborates the importance of the annotation step in GAL. We provide more analysis in Appendix A.3.

6 Limitations

This work demonstrates that one can leverage synthetic in-domain data generated by powerful pre-trained generative models. For simplicity, we do not employ any filtering avenue to retain diverse but high-quality data points. However, previous work has shown that advanced filtering approaches can further improve the performance (Sohn et al., 2020; Du et al., 2020; Yang et al., 2020). Given that the improvements in the self-training are not sizeable, we believe it is worth imposing filtering methods on the synthetic data to mitigate the side effects caused by the noisy data points.

Although we examine the effectiveness of GAL on various classification tasks, we still focus on the sentence-level tasks. Because of the superior performance on sentence-level tasks, there has been a surge of interest shift to document-level tasks, such as document-level machine translation (Miculicich et al., 2018; Voita et al., 2018; Maruf and Haffari, 2018), document summarization (Rush et al., 2015; Nallapati et al., 2016), and so forth. As these tasks suffer from data scarcity, one can leverage GAL to synthesize more data points. However, previous work has shown that GPT-2 has difficulty generating coherent text requiring long-range dependency (Orbach and Goldberg, 2020; Guan et al., 2020). Thus, such a limitation may hinder the application of GAL to document-level tasks.

In addition, the label space of the studied tasks is not as complex as the structured prediction tasks, such as machine translation, dialog system, question answering, and so on. However, we believe one can smoothly adapt GAL to these tasks as well. Let us consider machine translation (MT) as a canonical structured prediction task. Prior work has shown that one can use (real) monolingual data, in either source or the target language, through data augmentation (Sennrich et al., 2016) or knowledge distillation (Kim and Rush, 2016) to improve the structured prediction tasks. This suggests a promising avenue for future research on using synthetically generate monolingual data to improve MT for specialized domains where even monolingual data is scarce.

Generative model	Labeled synthetic data	SST-2	RTE	MRPC	CoLA
None (baseline)	–	94.8	78.8	90.1	63.6
Class-conditional LM	✓	92.9	74.4	86.0	58.4
Unconditional LM (GAL)	✗	95.3	81.4	91.7	65.1
Class-conditional LM (GAL)	✗	95.4	81.0	91.4	65.2

Table 7: Synthetic data from class-conditional LMs underperforms GAL and RoBERTa on GLUE dev sets.

Furthermore, Vu et al. (2021a) suggest that one can leverage a retrieval-based approach to obtain monolingual sentences from the generic data stores. This retrieved monolingual data is then employed to improve the translation quality in a domain adaptation setting. This suggests that a GAL-based approach to synthetically generate monolingual text is a promising method to improve MT for specialized domains—an interesting direction for future research.

7 Conclusion

We present Generate, Annotate, and Learn (GAL): a framework for self-training and knowledge distillation with generated unlabeled data. We motivate GAL from an expected risk minimization perspective and demonstrate both theoretically and empirically that the use of unconditional generative models for synthetic data generation is more effective than class-conditional generative models previously used in the literature. GAL leverages advances in large pretrained language models to help supervised learning and can have implications for learning from limited labeled data. GAL significantly helps improve knowledge distillation and prompt-based few-shot learning. In addition, a concurrent work (Gowal et al., 2021) has shown that using generated images can enhance the robustness of images classifiers. We will explore this direction on NLP tasks in the future. Finally, we hope that GAL will stimulate new research on the evaluation and development of large language models.

Acknowledgments

We would like to thank the anonymous reviewers and action editor André F.T. Martins for their comments and suggestions on this work. The computational resources of this work are partly supported by the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE) (www.massive.org.au). This material is partly based on research sponsored by Air Force Research Laboratory and DARPA under agreement number FA8750-19-2-0501. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

Steven Abney. 2004. Understanding the Yarowsky algorithm. *Computational Linguistics*, 30(3):365–395. https://doi.org/10.1162/0891201041850876

A. Agrawala. 1970. Learning with a probabilistic teacher. *IEEE Transactions on Information Theory*, 16(4):373–379. https://doi.org/10.1109/TIT.1970.1054472

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173. https://doi.org/10.18653/v1/P19-1620

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439. https://doi.org/10.1609/aaai.v34i05.6239

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter,

Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv:2005.14165*.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541. https://doi.org/10.1145/1150402.1150464

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C. Duchi, and Percy S. Liang. 2019. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. 2001. Vicinal risk minimization. *Advances in Neural Information Processing Systems*.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. 2020a. Big self-supervised models are strong semi-supervised learners. *NeurIPS*.

Yining Chen, Colin Wei, Ananya Kumar, and Tengyu Ma. 2020b. Self-training avoids using spurious features under domain shift. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *International Conference on Learning Representations*.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Ves Stoyanov, and Alexis Conneau. 2020. Self-training improves pre-training for natural language understanding. *arXiv:2010.02194*.

Jason Eisner and Damianos Karakos. 2005. Bootstrapping without the boot. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 395–402. https://doi.org/10.3115/1220575.1220625

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988.

S. Fralick. 1967. Learning to recognize patterns without a teacher. *IEEE Transactions on Information Theory*. https://doi.org/10.1109/TIT.1967.1053952

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. *International Conference on Machine Learning*, pages 1607–1616.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A. Mann. 2021. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108. https://doi.org/10.1162/tacl_a_00302

Gholamreza Haffari and Anoop Sarkar. 2007. Analysis of semi-supervised learning with the yarowsky algorithm. In *UAI 2007, Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, July 19-22, 2007*, pages 159–166. AUAI Press.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *arXiv:2006.03654*.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. 2021. Scaling laws for transfer.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv:1503.02531*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. TinyBERT: Distilling BERT for natural language understanding. *arXiv:1909.10351*. https://doi.org/10.18653/v1/2020.findings-emnlp.372

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. https://doi.org/10.18653/v1/D16-1139

Sosuke Kobayashi. 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. https://doi.org/10.18653/v1/N18-2072

Ananya Kumar, Tengyu Ma, and Percy Liang. 2020a. Understanding self-training for gradual domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5468–5479. PMLR.

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

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. *arXiv:1904.05780*. https://doi.org/10.18653/v1/N19-1333

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*.

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1275–1284. https://doi.org/10.18653/v1/P18-1118

Alana Marzoev, Samuel Madden, M. Frans Kaashoek, Michael J. Cafarella, and Jacob Andreas. 2020. Unnatural language processing: Bridging the gap between synthetic and natural language data. *ArXiv*, abs/2004.13645.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1325

Hossein Mobahi, Mehrdad Farajtabar, and Peter L. Bartlett. 2020. Self-distillation amplifies regularization in hilbert space. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290. https://doi.org/10.18653/v1/K16-1028

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319.

Sajad Norouzi, David J. Fleet, and Mohammad Norouzi. 2020. Exemplar vaes for exemplar based generation and data augmentation. *arXiv:2004.04795*.

Eyal Orbach and Yoav Goldberg. 2020. Facts2Story: Controlling text generation by key facts. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2329–2345, Barcelona, Spain (Online). International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.211

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. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53. https://doi.org/10.18653/v1/N19-4009

Samet Oymak and Talha Cihad Gulcu. 2020. Statistical and algorithmic insights for semi-supervised learning with self-training. *CoRR*, abs/2006.11006.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. https://d4mucfpksywv-models-models/language_models_are_unsupervised_multitask_learners.pdf

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. 2020. Understanding and mitigating the tradeoff between robustness and accuracy. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7909–7919. PMLR.

Ahmad Rashid, Vasileios Lioutas, and Mehdi Rezagholizadeh. 2021. Mate-kd: Masked adversarial text, a companion to knowledge distillation. *arXiv preprint arXiv:2105.05912*. https://doi.org/10.18653/v1/2021.acl-long.86

Suman Ravuri and Oriol Vinyals. 2019. Classification accuracy score for conditional generative models. *Advances in Neural Information Processing Systems*, pages 12268–12279.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

H. Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*. https://doi.org/10.1109/TIT.1965.1053799

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-1009

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*.

Sam Shleifer. 2019. Low resource text classification with ulmfit and backtranslation. *arXiv preprint arXiv:1903.09244*.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv:2001.07685*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019a. Patient knowledge distillation for BERT model compression. *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 4314–4323. https://doi.org/10.18653/v1/D19-1441

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019b. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association*

of Computational Linguistics, pages 25–32, Prague, Czech Republic. Association for Computational Linguistics.

Vladimir Vapnik. 1992. Principles of risk minimization for learning theory. *Advances in Neural Information Processing Systems*.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274. https://doi.org/10.18653/v1/P18-1117

Thuy Vu, Xuanli He, Dinh Phung, and Gholamreza Haffari. 2021a. Generalised unsupervised domain adaptation of neural machine translation with cross-lingual data selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3335–3346.

Tu Vu, Minh-Thang Luong, Quoc Le, Grady Simon, and Mohit Iyyer. 2021b. Strata: Self-training with task augmentation for better few-shot learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5715–5731. https://doi.org/10.18653/v1/2021.emnlp-main.462

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. 2019a. Can you tell me how to get past sesame street? Sentence-level pretraining beyond language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4465–4476. https://doi.org/10.18653/v1/P19-1439

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv: 1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019c. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *International Conference on Learning Representations*. https://doi.org/10.18653/v1/W18-5446

Bailin Wang, Wenpeng Yin, Xi Victoria Lin, and Caiming Xiong. 2021. Learning to synthesize data for semantic parsing. In *Proceedings of the Meeting of the North-American Chapter of Association for Computational Linguistics (NAACL)*. https://doi.org/10.18653/v1/2021.naacl-main.220

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 billion parameter autoregressive language model. https://github.com/kingoflolz/mesh-transformer-jax.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563. https://doi.org/10.18653/v1/D15-1306

Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2018. Kdgan: Knowledge distillation with generative adversarial networks. *NeurIPS*.

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1332–1342, Beijing, China. Association for Computational Linguistics. https://doi.org/10.3115/v1/P15-1129

Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. 2021. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*.

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 6382–6388. https://doi.org/10.18653/v1/D19-1670

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional BERT contextual augmentation. *International Conference on Computational Science*, pages 84–95, Springer. https://doi.org/10.1007/978-3-030-22747-0_7

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020. Self-training with noisy student improves imagenet classification. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695. https://doi.org/10.1109/CVPR42600.2020.01070

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. Bert-of-theseus: Compressing BERT by progressive module replacing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7859–7869.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. G-daug: Generative data augmentation for commonsense reasoning. *arXiv:2004.11546*. https://doi.org/10.18653/v1/2020.findings-emnlp.90

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196. https://doi.org/10.3115/981658.981684

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. QANet: Combining local convolution with global self-attention for reading comprehension. *ICLR*.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. *ICLR*.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722. https://doi.org/10.1109/ICCV.2019.00381

A Appendices

A.1 Datasets

The statistics of GLUE are reported in Table 8.

A.2 GPT-2 for Classification

We have conducted additional experiments, where we fine-tune GPT-2 as a classifier. We have considered two variants of the GPT-2 model. The first varant is the original GPT-2 model (GPT2-original) pre-trained on open-domain text. The second variant is the GPT-2 model that was fine-tuned on the inputs of each task separately (GPT-2-finetuned). This model was used to generate task-specific (synthetic) unlabeled data. Finally, we also consider self-training with GAL on top of GPT2-original. Specifically, we use the GPT-2-finetuned model to synthesize 40x in-domain unlabeled data. Then we apply self-training to GPT-2-original, where the data is a combination of the original labeled data and pseudo-labeled synthetic data. Table 9 suggests that the gains of GAL come from the pseudo-labeled synthetic data, i.e., both synthetic unlabeled data and teacher's knowledge. Without the generation of synthetic unlabeled data, the domain-specific knowledge embedded in GPT-2-finetuned model cannot be utilized. As such, GPT-2-finetuned model is inferior to the GPT2-original model. Since RoBERTa-large is superior to GPT-2 models, RoBERTa-large+GAL also significantly outperform the GPT-2 counterpart.

A.3 Importance of Pseudo-labels

We have argued and demonstrated that using class-conditional generative models to generate

Dataset	task	domain	#train	#dev	#test	#classes
SST-2	sentiment analysis	movie reviews	67k	872	1.8k	2
QQP	paraphrase	social QA questions	364k	40k	391k	2
QNLI	QA/natural language inference	Wikipedia	105k	5k	5.4k	2
RTE	natural language inference	news, Wikipedia	2.5k	277	3k	2
MNLI	natural language inference	misc.	393k	20k	20k	3
MRPC	paraphrase	news	3.7k	408	1.7k	2
CoLA	acceptability	misc.	8.5k	1043	1k	2
STS-B	sentence similarity	misc.	5.8k	15k	1.4k	–

Table 8: Summary of the three sets of tasks used for evaluation of GAL. STS-B is a regression task, so #classes is not applicable.

Model	MNLI	CoLA	SST-2	MRPC	STS-B	QQP	QNLI	RTE	Avg
GPT-2-original	85.9/85.6	54.8	94.5	86.9/82.2	86.3/85.2	72.5/89.3	91.2	69.8	80.9
GPT-2-finetuned	85.8/85.5	40.9	94.5	87.0/81.0	85.6/84.3	71.4/88.5	91.5	69.0	78.8
GPT-2-original+GAL	86.2/85.8	55.7	94.7	87.9/83.4	86.9/85.9	72.6/89.4	91.9	70.6	81.5
RoBERTa-large	90.1/89.7	63.8	96.1	91.2/88.3	90.9/90.7	72.5/89.6	94.5	85.9	86.5
RoBERTa-large + GAL	90.2/89.8	66.2	96.4	92.0/89.2	90.7/90.5	73.6/89.9	95.0	86.3	87.1

Table 9: GLUE test results of using GPT-2 and RoBERTa-large as classification models.

Label type	Accuracy	F1	Precision	Recall
GPT2	86.0	87.0	88.7	85.5
RoBERTa	90.0	91.4	100.0	84.1
conditioning label	72.0	71.4	66.0	77.8

Table 10: Performance of GPT2 annotation, RoBERTa annotation and conditioning labels on 100 random examples from the synthetic RTE dataset generated by a class-conditional LM.

labeled synthetic examples is less effective than GAL in Section 3 and Section 5. To further verify this argument, we sample 100 instances from the synthetic RTE dataset generated by the label-prompted GPT2, as the class-conditional LM. Then we annotate these examples using a human annotator, GPT2 classifier, and RoBERTa classifier. Finally, we compute the Accuracy, F1, Precision, and Recall scores between human labels and GPT2 labels, between human labels and RoBERTa labels, and between human labels and conditioned labels used by GPT2 when the data was generated. Table 10 shows that class-conditional LM has difficulty generating sentences retaining the semantics or pragmatics of a specified category, which also corroborates our theoretical analysis in Section 3. On the other hand, discriminative models, such as GPT2 classifier and RoBERTa classifier, are able to produce higher quality labels that correlate better with human annotations.

A.4 Generated Unlabeled Examples Annotated with Pseudo Labels

We provide some synthetic sentences generated by GAL in Tables 11 and 12.

are more deeply thought through than in most ‘ right-thinking ’ films (**positive**)
KNN: 1: is far more sophisticated, insightful and thought-provoking than his previous films . (**positive**) 2: is more sophisticated than its more obvious and less-than-dazzling counterparts (**positive**) 3: is about as well-thought as the idea of a bad hair day, (**negative**)
contains no wit, only labored gags (**negative**)
KNN: 1: lacks insight, and lacks empathy (**negative**) 2: has little humor or intelligence (**negative**) 3: lacks all wit and humanity (**negative**)

Table 11: **SST-2**: Two labeled examples, along with 3 nearest neighbors (based on RoBERTa representations) from our synthetic dataset. We include **labels** for original examples and **pseudo-labels** for synthetic examples in parenthesis.

How is the life of a math student? Could you describe your own experiences? [SEP] Which level of prepration is enough for the exam jlpt5? (**not duplicated**)
KNN: 1: What are the best courses for a mechanical engineering student? [SEP] What is the best course to do after completing a B.Tech in mechanical engineering? (**not duplicated**) 2: How much marks are needed to get through the GATE with electronics? [SEP] What is the average score of the Gate EE exam? What are the cut-offs? (**not duplicated**) 3: What is the best time table for students to prepare for IAS? [SEP] How can one study for IAS in a best time? (**not duplicated**)
How does an IQ test work and what is determined from an IQ test? [SEP] How does IQ test works? (**duplicated**)
KNN: 1: What is the average IQ of the U.S. population? [SEP] How does an IQ test work? (**not duplicated**) 2: Is the Iq test an effective way to measure intelligence? [SEP] How do IQ tests work? (**duplicated**) 3: How is an IQ test on a scale from 1 to 100 scored? [SEP] How do you get your IQ tested? (**not duplicated**)

Table 12: **QQP**: Two labeled examples, along with 3 nearest neighbors (based on RoBERTa representations) from our synthetic dataset. We include **labels** for original examples and **pseudo-labels** for synthetic examples in parenthesis.