KEST: Kernel Distance Based Efficient Self-Training for Improving Controllable Text Generation

Yuxi Feng[1] , **Xiaoyuan Yi**[2*] , **Laks V.S. Lakshmanan**[1] and **Xing Xie**[2]
[1]The University of British Columbia, Vancouver, Canada
[2]Microsoft Research Asia, Beijing, China
{fyx14,laks}@cs.ubc.ca, {xiaoyuanyi, xing.xie}@microsoft.com

Abstract

Self-training (ST) has come to fruition in language understanding tasks by producing pseudo labels, which reduces the labeling bottleneck of language model fine-tuning. Nevertheless, in facilitating semi-supervised controllable language generation, ST faces two key challenges. First, augmented by self-generated pseudo text, generation models tend to over-exploit the previously learned text distribution, suffering from mode collapse and poor generation diversity. Second, generating pseudo text in each iteration is time-consuming, severely decelerating the training process. In this work, we propose KEST, a novel and efficient self-training framework to handle these problems. KEST utilizes a kernel-based loss, rather than standard cross entropy, to learn from the soft pseudo text produced by a shared non-autoregressive generator. We demonstrate both theoretically and empirically that KEST can benefit from more diverse pseudo text in an efficient manner, which allows not only refining and exploiting the previously fitted distribution but also enhanced exploration towards a larger potential text space, providing a guarantee of improved performance. Experiments on three controllable generation tasks demonstrate that KEST significantly improves control accuracy while maintaining comparable text fluency and generation diversity against several strong baselines.

1 Introduction

Recent years have witnessed the excellence of Pretrained Language Models (PLMs) [Liu *et al.*, 2019; Dong *et al.*, 2019; Radford *et al.*, 2019; Raffel *et al.*, 2020] in Natural Language Processing (NLP). However, these PLMs still rely on increasingly more labeled instances for fine-tuning with growing model size [Yogatama *et al.*, 2019], hampering their effectiveness under insufficient data [Zhang *et al.*, 2021]. To solve this problem, a promising approach is *Self-training (ST)* [Scudder, 1965; Yarowsky, 1995; Grandvalet and Bengio, 2004], a classic semi-supervised learning [Chapelle *et al.*, 2006] paradigm. ST minimizes the prohibitively expensive human labeling by iteratively pseudo-annotating unlabeled data with a classifier which is then retrained with the augmented labels. In this way, ST benefits from a vast number of unlabeled instances and extends the generalization bound [Wei *et al.*, 2021b; Zhang *et al.*, 2022], boosting a wide spectrum of tasks like Image Classification [Han *et al.*, 2019; Xie *et al.*, 2020], Speech Recognition [Park *et al.*, 2020], and Natural Language Understanding (NLU) [Mukherjee and Hassan Awadallah, 2020; Vu *et al.*, 2021; Li *et al.*, 2021].

Nonetheless, it is unresolved how to incorporate ST into the data-intensive attribute-controllable Natural Language Generation (NLG), i.e., generate a textual sequence satisfying the input attribute label, as opposed to NLU. Since model inputs now are discrete labels, massive high-quality unlabeled target text (*e.g.*, movie reviews for sentiment-controllable NLG) is essential to construct pseudo label-text pairs, which is impractical in low-resource domains, impeding the broad application of ST [Du *et al.*, 2021]. Consequently, classical ST only works for a few generation tasks with adequate plain text, like Sequence Labeling [Wang *et al.*, 2020] and Machine Translation [He *et al.*, 2020; Jiao *et al.*, 2021].

With limited unlabeled text, a potential approach to further improve ST performance is to leverage the generative ability of NLG models and produce synthetic (pseudo) text [Yang *et al.*, 2020; Schick and Schütze, 2021] from given labels besides pseudo labels from text. In this case, unfortunately, two major challenges arise. **i) Over-exploitation**: Augmented by self-generated text, NLG models are forced to repeatedly fit the already learned text distribution. This gradually homogenizes the generated pseudo text and causes a shrunken (collapsed) generalization boundary, resulting in decreased controllability and generation diversity. **ii) Training deceleration**: We need to re-generate all pseudo text in each ST iteration with updated model parameters, which interrupts the parallelism of Transformer [Vaswani *et al.*, 2017]-based models, severely decelerating training and impairing practicality.

To tackle these challenges, we propose a novel self-training framework, **K**ernel Distance Based **E**fficient **S**elf **T**raining (**KEST**), for improving semi-supervised controllable NLG. Instead of learning from generated pseudo textual sequences with traditional cross-entropy loss, KEST directly fits the ap-

*Work done during Yuxi Feng's internship at Microsoft Research Asia mentored by Xiaoyuan Yi.

proximated text distribution from the last iteration in the embedding space. Such an objective not only relaxes the constraint imposed by the previous ST iteration but also encourages diverse outputs of the current model, addressing *Challenge (i)*. Besides, we design a non-autoregressive generation schema to produce soft representations of pseudo text (rather than hard strings) in parallel, greatly reducing time cost and handling *Challenge (ii)*. Furthermore, such a soft text is naturally a kind of noisy pseudo data [He *et al.*, 2020; Xie *et al.*, 2020], which helps the model denoise errors and propagate local smoothness [Wei *et al.*, 2021b; Chen *et al.*, 2021]. Our method can be theoretically interpreted as exploring a larger potential text space, leading to an extended generalization boundary and improved controllability while maintaining comparable generation quality. [1]

In summary, our contributions are as follows:

- We dig into the over-exploitation problem of applying self-training to controllable NLG and propose a novel kernel-based ST framework to address this problem.
- We design a non-autoregressive generation schema to reduce the time cost of producing pseudo text for self-training, making ST more practical for real scenarios.
- We theoretically show that KEST could explore a larger potential text space and demonstrate through exhaustive experiments that our model significantly improves controllability with competitive generation diversity and quality, further exploring the capacity frontier of PLMs.

2 Related Work

Controllable Natural Language Generation. Attribute-controllable NLG seeks to make the generated text satisfy user-specified attributes *e.g.*, sentiment, topic and style [Keskar *et al.*, 2019; Dathathri *et al.*, 2020] while keeping satisfactory generation quality, which could benefit various downstream applications. With the impressive generation ability of PLMs, a common practice for controllable NLG is fine-tuning a PLM conditioned on attribute labels with attribute-text paris [Keskar *et al.*, 2019; Gururangan *et al.*, 2020]. However, as the scale of PLMs keeps increasing, insufficient labeled data becomes a new obstacle to fine-tuning [Yogatama *et al.*, 2019; Zhang *et al.*, 2021]. To alleviate this problem, another line of methods, called *Plug-in Control*, has been established, which manipulates the output generation probability of models to encourage attribute-related tokens. The manipulation is achieved broadly through two paradigms: updating cached hidden states [Dathathri *et al.*, 2020] or reshaping the output distribution guided by off-the-shelf attribute classifiers [Krause *et al.*, 2021; Yang *et al.*, 2023] or conditional PLMs [Liu *et al.*, 2021] at inference time without fine-tuning. Despite reduced labeling costs, with weak/sparse attribute signals, these methods usually hurt control accuracy or generation fluency.

Self-training. Self-training (ST) [Yarowsky, 1995; Grandvalet and Bengio, 2004] has recently found renewed interest and exhibited notable advantages of augmenting PLM fine-tuning. This paradigm iteratively produces pseudo labels for massive unlabeled data and reduces labeling bottleneck, facilitating varied downstream tasks where massive unlabeled in-domain text exists, including NLU [Vu *et al.*, 2021; Du *et al.*, 2021; Bhat *et al.*, 2021; Chen *et al.*, 2021], Image Classification [Han *et al.*, 2019; Xie *et al.*, 2020; Sohn *et al.*, 2020], Speech Recognition [Park *et al.*, 2020; Kahn *et al.*, 2020], and Neural Machine Translation (NMT) [Zhang and Zong, 2016; He *et al.*, 2020; Jiao *et al.*, 2021]. Besides classical ST, diverse follow-up modifications have been developed for further improvement, which generally fall into two lines. The first line, *i.e.*, sample selection, selects only a part of unlabeled instances in terms of (1) model confidence to avoid over-noisy pseudo labels [Sohn *et al.*, 2020; Bhat *et al.*, 2021], (2) prediction uncertainty to obtain informative instances and enhance performance on the hard ones [Mukherjee and Hassan Awadallah, 2020; Jiao *et al.*, 2021], or (3) label balance to benefit minority classes [Wei *et al.*, 2021a]. The other line is noisy labeling [He *et al.*, 2020; Xie *et al.*, 2020], which injects synthetic noise into the pseudo data, *e.g.*, token shuffle or image distortion to propagate local smoothness and improve model robustness. Despite remarkable progress, as discussed in Sec.1, these ST methods are unsuitable for attribute-controllable NLG because of the two challenges identified earlier.

Non-Autoregressive Generation (NAG). Relevant to our work, NAG aims to simultaneously generate all target tokens rather than one by one to increase the inference speed. NAG was first proposed in NMT [Gu *et al.*, 2018; Ma *et al.*, 2019] and then applied to broader scenarios like Text Summarization [Liu *et al.*, 2022] and Text-to-Speech Synthesis [Chien and Lee, 2021]. All the tasks are learned with encoder-decoder architectures, relying on long input sequences (*e.g.*, source language) to provide rich initial context information. However, it is still challenging to leverage NAG for our task since the inputs are only attribute labels and short prompts.

Unlike all the works mentioned above, we take a further step to investigate the challenges of incorporating ST with controllable NLG and propose a practical NAG method to generate soft pseudo text, which is then learned in a kernel space, leading to a novel and efficient ST framework.

3 Method

3.1 Formulation and Overview

Let $\mathbf{x}_i$ denote a textual sequence and y_i an attribute label. Assume we have a labeled dataset $D_l = \{\mathbf{x}_i, y_i\}_{i=1}^{N_l}$, and an unlabeled in-domain set $D_u = \{\mathbf{x}_i\}_{i=1}^{N_u}$ where $N_u \gg N_l$. Our goal is to learn an attribute-controllable generator $\mathcal{G}_{ag}(y) = P_\theta(\mathbf{x}|y)$ (parameterized by θ) to generate high-quality text $\mathbf{x}$, matching the given label y. In addition, we endow the generator with the ability of multi-task generation. Concretely, the model is reused and jointly trained to generate (a) pseudo text $\hat{\mathbf{x}}$ in a non-autoregressive manner, depicted as $\mathcal{G}_{nag}(y)$, for further augmenting self-training, and (b) pseudo labels $\hat{y}$ for $\mathbf{x} \in D_u$, namely, a classifier $\mathcal{C} = P_\theta(y|\mathbf{x})$.

During the self-training phase, besides the pseudo label pairs $(\mathbf{x}, \hat{y})$, KEST also learns the pseudo text pairs $(\hat{\mathbf{x}}, y)$

[1] Code and appendices are available at https://github.com/peterfengyx/KEST.

in the kernel space to simultaneously cover more unseen instances and extend the previously fitted distribution (Sec.3.4), *handling Challenge (i)*. All the pseudo text $\hat{\mathbf{x}}$ is produced through NAG efficiently, *handling Challenge (ii)*.

3.2 Multi-task Generator

To further enhance the performance and efficiency of our model, we design a multi-task generator to produce the desired text $\mathbf{x}$, *Pseudo Label (PL)* $\hat{y}$, and *Pseudo Text (PT)* $\hat{\mathbf{x}}$ jointly based on a shared PLM.

Autoregressive Text Generation. To obtain high-quality attribute-specified generated text $\mathbf{x}$, we optimize the generator $\mathcal{G}_{ag}$ in an autoregressive manner as follows:

$$\mathcal{L}_{ag} = -\frac{1}{N} \sum_{(\mathbf{x},y)\in D} [\sum_{j=1}^{L} \log P_\theta(\mathbf{x}^j|\mathbf{x}^{<j}, y)], \tag{1}$$

where $\mathbf{x}^j$ means the j-th token in $\mathbf{x}$, L is the length of $\mathbf{x}$, D is the training set with N samples. We will show later how to construct D for different training phases.

Pseudo Label Generation. We also make our model simultaneously learn a classifier $\mathcal{C}$ by minimizing:

$$\mathcal{L}_c = -\frac{1}{N} \sum_{(\mathbf{x},y)\in D} \log P_\theta(y|\mathbf{x}). \tag{2}$$

Eq. (2) enables our model to make full use of available unlabled text $\mathbf{x} \in D_u$ to produce pseudo labels by $\hat{y} = \mathrm{MLP}\,(\mathrm{Encoder}(\mathbf{x}))$, helping regularize the training and improve the generalization bound [Wei *et al.*, 2021b].

Non-autoregressive Pseudo Text Generation. With insufficient unlabeled text, we could produce pseudo text for further improvement and then speed up the repetitive PT generation via NAG. However, as shown in Sec. 2, an input consisting of just y is too uninformative to guide the generation, hampering convergence and causing extremely noisy PT. To mitigate this problem, we resort to the Masked Language Model (MLM) [Devlin *et al.*, 2019] to train the NAG generator $\mathcal{G}_{nag}$ and conduct generation. Define $\mathbf{m} \sim \mathcal{B}(L, p_m)$ as a mask indicator vector, where $\mathcal{B}$ is the Bernoulli distribution. Given a text $\mathbf{x}$, we replace part of the tokens in it with the MASK symbol and get the masked one $\mathbf{x}^{\backslash\mathbf{m}} = [\mathbf{x}^1, \cdots, \mathrm{MASK}, \cdots, \mathbf{x}^L]$, where $\mathbf{x}^j = \mathrm{MASK}$ iff $\mathbf{m}^j = 1$. Then we optimize the following loss for NAG:

$$\mathcal{L}_{nag} = -\frac{1}{N} \sum_{(\mathbf{x},y)\in D} [\sum_{j=1}^{L} \mathbb{I}(\mathbf{m}_j = 1) \log P_\theta(\mathbf{x}^j|\mathbf{x}^{\backslash\mathbf{m}}, y)], \tag{3}$$

where $\mathbb{I}$ is the indicator function and the masking probability p_m can be adjusted as the noise level. In this way, our model only needs to predict partial tokens according to the rich context $\mathbf{x}^{\backslash\mathbf{m}}$, which is easier to learn, reducing the time complexity of PT generation from $\mathcal{O}(L)$ to $\mathcal{O}(1)$ (see Fig. 2). Besides, the pseudo text $\hat{\mathbf{x}} = \mathcal{G}_{nag}(\mathbf{x}^{\backslash\mathbf{m}}, y)$ naturally introduces moderate noise in terms of re-predicted tokens while maintaining satisfactory fluency due to the unaltered high-quality ones. Such a flexible corruption acts as a kind of weak augmentation [Chen *et al.*, 2021] which enhances the exploitation and outperforms typical synthetic noise (*e.g.*, token dropout) [He *et al.*, 2020].

The final loss is computed as follows:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_{ag}\mathcal{L}_{ag} + \lambda_{nag}\mathcal{L}_{nag} \tag{4}$$

where λ_c, λ_{ag}, and λ_{nag} are hyper-parameters.

3.3 Kernel-based Learning

As we discussed in Sec. 1, learning from self-generated pseudo text $\hat{\mathbf{x}}$ with the standard cross-entropy loss forces the current model P_θ to over-exploit and is shackled to the previously learned one $P_{\theta'}$ (Sec. 3.4), resulting in a shrunken generalization boundary and decreased controllability.

To break such constraints, we make the current model P_θ directly fit the previous one $P_{\theta'}$. For this goal, we leverage *Maximum Mean Discrepancy (MMD)* [Gretton *et al.*, 2012], a well-known kernel-based probability measure, and minimize the following empirical loss for all generated pseudo text:

$$\begin{aligned}\mathcal{L}_{ker} =& \frac{1}{N(N-1)} \sum_{\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \in D_o, i\neq j} k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) \\ &- \frac{2}{N^2} \sum_{\tilde{\mathbf{x}}_i \in D_o, \hat{\mathbf{x}}_j \in D_{pt}} k(\tilde{\mathbf{x}}_i, \hat{\mathbf{x}}_j),\end{aligned} \tag{5}$$

where $D_{pt} = \{\hat{\mathbf{x}}_i\}_{i=1}^N$ is set of pseudo text, $D_o = \{\tilde{\mathbf{x}}_i\}_{i=1}^N$ is set of text generated by $\mathcal{G}_{ag}(\hat{\mathbf{x}}_i, y)$ (or $\mathcal{G}_{nag}(\hat{\mathbf{x}}_i^{\backslash\mathbf{m}}, y)$) in the self-training phase. k is the kernel function, for which we take the RBF kernel here, that is, $k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) = \exp\left(\frac{-\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|^2}{2\sigma^2}\right)$ and σ is the bandwidth.

This MMD loss is an unbiased estimator and model parameters can be learned through back-propagation. We will demonstrate in Sec. 3.4 that such an objective could relax the constraint imposed by the previous model $P_{\theta'}$ and encourage more diverse outputs.

Soft Pseudo Text (SPT). When optimizing Eq. (5), we need to calculate the l_2-distance between two text $\mathbf{x}_i$ and $\mathbf{x}_j$. Simply using hard text (one-hot representations) has two drawbacks. First, the signal would be too sparse since most dimensions are zeros in the vector. Second, the sampled discrete $\mathbf{x}_i$ (a point in the text space) causes information loss and forces us to sample numerous points to cover a small neighborhood region in the space. Therefore, we further propose to generate soft pseudo text. We use the feature representation of the text $\mathbf{x}$, $e(\mathbf{x}) = P(\mathbf{x}) \times \mathbf{E} \in \mathbb{R}^{L\times d}$, where $P(\mathbf{x}) \in \mathbb{R}^{L\times V}$ are the generation probabilities of each token $\mathbf{x}^i$ on the vocabulary, and $\mathbf{E} \in \mathbb{R}^{V\times d}$ is the word embedding matrix. V and d are vocabulary and embedding sizes, respectively. Then we change Eq.(1) and Eq. (3) to:

$$\begin{aligned}\mathcal{L}'_{ag} &= \mathcal{L}_{ker} \text{ if } \mathbf{x} \in D_{pt} \text{ else } \mathcal{L}_{ag} \\ \mathcal{L}'_{nag} &= \mathcal{L}_{ker} \text{ if } \mathbf{x} \in D_{pt} \text{ else } \mathcal{L}_{nag}.\end{aligned} \tag{6}$$

In this way, we avoid losing relevant semantics information in the pseudo text, make the model fit a smoother distribution and further extend the generalization boundary (see Table 3).

Algorithm 1 Training Process of KEST

Input: Labeled set D_l, unlabeled set D_u
Output: The trained model P_θ

```
Jointly train base model G_ag, G_nag, C on D_l by optimizing Eq.(4), store the best G_ag^0, G_nag^0, C^0.
for epoch ← 1 to MaxEpoch do
    for x_i in D_u do
        ŷ_i = C^{epoch-1}(x_i)
    end for
    Build pseudo labeled dataset D_pl = {x_i, ŷ_i}
    Sample a subset D_pseudo from D_l ∪ D_pl
    for (x_i, y_i) in D_pseudo do
        Sample mask vector m.
        x̂_i = G_nag^{epoch-1}(x_i^{\m}, y_i)
    end for
    Build pseudo text dataset: D_pt = {x̂_i, y_i}
    Train G_ag^{epoch-1}, G_nag^{epoch-1}, and C_{epoch-1} on {D_pt, D_pl, D_l} by optimizing Eq.(4) and Eq.(6), update the parameter to G_ag^{epoch}, G_ag^{epoch}, and C^{epoch}.
end for
```

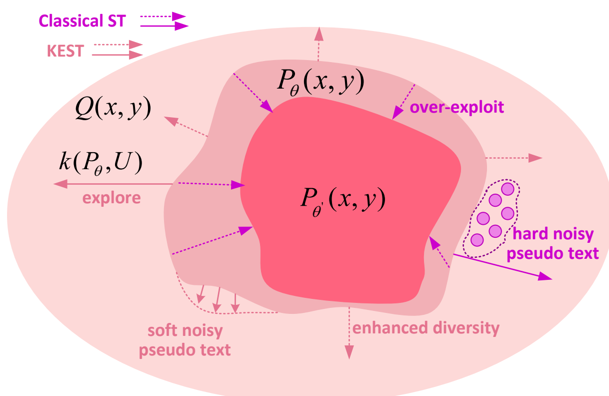

Figure 1: The illustration of KEST advantages.

Following the practice of self-training in NLU [Vu *et al.*, 2021], we start ST from a strong base model tuned on D_l and use the full unlabeled D_u to produce pseudo labels, rather than select part of the data with certain criteria as in [Mukherjee and Hassan Awadallah, 2020; Jiao *et al.*, 2021]. The PLM word embedding $\mathbf{E}$ is frozen during self-training. The complete KEST process is described in Alg. 1.

3.4 Further Analysis of KEST

To better understand the advantages of KEST, we provide the following two results.

Lemma 1. *The optimization of classical self-training is equivalent to minimizing* $(1-\alpha)*KL[Q(x,y)||P_\theta(x,y)] + \alpha * KL[P_{\theta'}(x,y)||P_\theta(x,y)]$, *where Q is the real joint distribution of text and label, P_θ and $P_{\theta'}$ are models estimated at the current and last ST iteration, respectively, KL is the Kullback–Leibler divergence, and α is the ratio of pseudo text.*

From Lemma 1, we can see that classical ST approximates the text distribution and fits the current model into the previously learned one. Since $\text{KL}[P_{\theta'}(x,y)||P_\theta(x,y)] = \iint P_{\theta'}(x,y)\log\frac{P_{\theta'}(x,y)}{P_\theta(x,y)}dxdy$, failing to assign enough probability mass to a point (x,y) in $P_{\theta'}$ will bring extremely larger loss. Consequently, P_θ is more inclined to cover $P_{\theta'}$ rather than explore Q, causing over-exploitation.

In contrast, we give a theorem of our KEST:

Theorem 1. *Minimizing the training objective of KEST is equivalent to minimizing the following:*

$$\begin{aligned} &KL\left[Q(x,y)||P_\theta(x,y)\right] \\ &+MMD^2\left[P_{\theta'}(x,y)||P_\theta(x,y)\right] \\ &-2*\mathbb{E}_{P_\theta U}\left[k(x,u)\right], \end{aligned} \tag{7}$$

where U is a noise distribution.

Proof. See Appendix B.

In Theorem 1, our KEST fits the true distribution Q by KL divergence to cover the real space as large as possible while fitting $P_{\theta'}$ with MMD. Considering Eq.(5), we can see this loss not only regularizes P_θ by $P_{\theta'}$, but also diversifies P_θ via increasing the l_2-distance of generated text $\|\mathbf{x}_i - \mathbf{x}_j\|^2$, and enhances exploration through fitting a noise distribution and disturbing P_θ, further pushing the generalization boundary.

4 Experiments

4.1 Tasks

We conduct exhaustive experiments on three controllable generation tasks, described below:

Sentiment control with prompt. We evaluate the sentiment controllability on the IMDb movie review dataset [Maas *et al.*, 2011]. Following [Dathathri *et al.*, 2020], we use their 15 prompts and another 85 prompts sampled from IMDb (100 in total) as model input. We generate 10 samples for each prompt and each sentiment.

Topic control w/o prompt. We use the AGNews dataset [Zhang *et al.*, 2015] to evaluate topic controllability. We assess our model's ability to generate from scratch on this dataset and sample 300 generations for each topic.

Text detoxification. We use the Jigsaw Toxicity Dataset for training, and use the 203 "challenging" prompts (toxicity$<$ 0.5) from [Gehman *et al.*, 2020] to generate 10 non-toxic sentences for each prompt, following [Qian *et al.*, 2022].

For IMDb, we sample 5% of the training samples as labeled data and directly take their provided unlabeled set. Since there is no separate unlabeled text in AGNews, we sample 3% of training samples as labeled data and use the others as unlabeled ones. For a fair comparison, we keep the ratio of labeled/pseudo/unlabeled text to 1:1:30. More details of the dataset are provided in Appendix A.2.

4.2 Experimental Settings

We use UniLM-base-cased [Dong *et al.*, 2019] as the shared classifier and generator. We use AdamW [Loshchilov and Hutter, 2019] with learning rate = 5e-5, warm-up steps = one epoch, and batch size = 8 for optimization. The top-p (p = 0.9) sampling method is used for decoding in evaluation. We set $\lambda_c = \lambda_{ag} = \lambda_{nag} = 1.0$ in Eq. (4) across all tasks. More implementation details are provided in Appendix A.1.

	Sentiment					Topic				
	O-PPL ↓	M-PPL ↓	F1 ↑	Dist ↑	S-BLEU ↓	O-PPL ↓	M-PPL ↓	F1 ↑	Dist ↑	S-BLEU ↓
Test set	25.14	–	96.20	48.27	43.34	31.04	–	94.89	67.24	23.31
GPT2 (raw)	13.20	38.39	68.50	35.91	58.79	16.94	74.41	52.17	46.88	45.55
Fine-tuned PLM										
GPT2	**16.40**	44.02	80.44	26.34	71.00	**22.22**	23.46	83.08	54.93	<u>39.93</u>
UniLM	25.20	54.33	75.35	31.05	66.97	55.79	36.28	87.70	54.76	43.77
T5	25.69	34.97	83.77	30.03	69.57	48.33	32.12	88.43	**58.06**	**37.01**
Self-Training methods										
PT	26.62	58.37	70.27	31.17	66.69	57.40	40.95	86.36	52.35	46.41
PT(noise)	30.28	62.07	75.78	<u>31.68</u>	**65.18**	58.59	45.32	85.27	53.35	46.57
PT(noise)+PL	18.92	**33.53**	89.73	30.94	66.84	32.36	**16.64**	89.70	53.79	47.95
PT(select)+PL	<u>18.40</u>	<u>33.56</u>	<u>90.06</u>	31.27	67.61	33.23	<u>16.66</u>	<u>90.52</u>	53.71	47.69
KEST	20.65	38.15	**91.77**	**31.70**	<u>66.60</u>	<u>31.19</u>	20.46	**91.94**	<u>56.16</u>	42.10

Table 1: Automatic evaluation results on IMDb dataset (sentiment) and AGNews dataset (topic). For each metric, the best results are in **bold**, and the second-best results are <u>underlined</u>.

4.3 Evaluation Metrics

We mainly focus on improving control accuracy and diversity while maintaining the generation quality in this work, considering the following four kinds of metrics. Due to the page limit, we also provide the classification performance of KEST in Appendix C.1.

Fluency. We evaluate generation fluency by the perplexity of generated text measured by a GPT2-XL [Radford *et al.*, 2019] model, *i.e.*, **Output PPL**.

Generalizability. We calculate the perplexity of each model on each held-out test set provided in each dataset, *i.e.*, **Model PPL**, which measures how well the model generalizes and adapts to the unseen domain under a specified attribute.

Controllability. We evaluate the control accuracy through classification Macro-F1 (**F1**) on the generated text by two RoBERTa-large classifiers fine-tuned on the full IMDb and AGNews data (testing F1=96% and 95%), respectively. For toxicity evaluation, we use the Perspective API[2].

Diversity. To evaluate the diversity of generated text, we consider **Dist-n** [Li *et al.*, 2016] and **Self-BLEU** [Zhu *et al.*, 2018].

More metrics details are described in Appendix A.3.

4.4 Baselines

We compare our model with the following (supervised or semi-supervised) strong NLG baselines.

Fine-tuned PLM. We fine-tune diverse powerful PLMs on each of the datasets, including GPT2 [Radford *et al.*, 2019], UniLM [Dong *et al.*, 2019] and T5 [Raffel *et al.*, 2020].

Self-training methods. (1) PT: the naive Self-training [Grandvalet and Bengio, 2004], which generates pseudo text at each epoch and updates parameters with both real and pseudo text. (2) PT(noise): the noisy version of Self-training [He *et al.*, 2020], which brings synthetic noise (token drop, swap and mask) to the pseudo text for self-training. (3) PT(noise)+PL: We combine PT(noise) with *pseudo labeling*, and fine-tune a BERT-base [Devlin *et al.*, 2019] on D_l to generate pseudo labels for all real unlabeled text. (4) PT(select)+PL: The sample selection version of ST [Mukherjee and Hassan Awadallah, 2020]. We over-generate noisy pseudo text and select samples by the classifier confidence and uncertainty scores. All the self-training methods are applied to the same fine-tuned UniLM as used in KEST.

We give more details of the baseline models above in Appendix A.4.

4.5 Results

As shown in Table 1, in both sentiment (with prompt) and topic (without prompt) controlled generation, our KEST achieves significant improvement in control accuracy (+8.0 F1 at most) compared to fine-tuned PLMs. The generally much higher PPL (for UniLM and T5), limited F1 improvement, and severely decreased diversity (for GPT2) indicate these PLMs either fail to be adapted to new domains (*e.g.*, positive movie reviews) or overfit with inadequate labeled data, as analyzed in [Zhang *et al.*, 2021]. On the contrary, thanks to the self-augmented data, KEST notably enhances controllability as well as fluency and diversity, especially compared to the backbone UniLM.

We also observed some interesting results considering existing self-training methods. 1) The naive self-training with PT performed poorly in controllability and diversity, even worse than tuned PLMs, due to the over-exploitation and shrunken distributions as interpreted in Sec. 3.4. 2) The traditional synthetic noise (PT(noise)) slightly boosts control accuracy and diversity, which verifies the effectiveness of noise [He *et al.*, 2020] again, but greatly hurts fluency and generalizability (+3.7 O-PPL at most). This is because such hard corruption is too noisy and makes the model diverge far from valid attribute distributions. In contrast, KEST utilizes a NAG generator to produce flexible noise, improving local smoothness. 3) Additional pseudo-labels bring significant improvement, especially on PPL. However, with a fixed number of unlabeled data, the performance of these methods

[2]https://www.perspectiveapi.com/

	Fluency ↑	Novelty ↑	Rel. ↑
Sentiment			
UniLM-PT(select)+PL	3.60	3.40	3.62**
KEST	**3.67**	**3.48**	**3.87**
Topic			
UniLM-PT(select)+PL	3.97**	4.43	4.57
KEST	**4.11**	**4.54**	**4.62**

Table 2: Human evaluation results on sentiment/topic-controlled generation. We conduct the Student t-test to evaluate statistical significance (**: p-value$<$ 0.01). The overall Cohen's kappa score is 0.62, showing a satisfactory inter-annotator agreement.

	AGNews			
	O-PPL	M-PPL	F1	S-BLEU
KEST	**31.19**	**20.46**	**91.94**	**42.10**
−Soft	38.04	29.07	90.96	44.09
$-\mathcal{L}_{ker}$−Soft	38.98	28.77	90.81	45.02
$-\mathcal{L}_{nag}-\mathcal{L}_{ker}$−Soft	39.73	28.58	90.42	44.73
−PT	38.09	28.77	90.97	44.13
−PL−PT	37.24	256.66	87.45	69.30

Table 3: Ablation study on AGNews dataset. The symbol − means removing the settings from KEST. −Soft: using sampled hard tokens instead of the soft $e(\mathbf{x})$. $-\mathcal{L}_{ker}$: using the cross-entropy loss instead of Eq.(5). $-\mathcal{L}_{nag}$: using $\mathcal{G}_{ag}$ to generate pseudo text instead of $\mathcal{G}_{nag}$. −PT/−PL: do not use pseudo text/labels.

is limited. Besides, KEST utilizes the multi-task generator to produce soft pseudo text in feature space, which helps cover a larger attribute space and obtain further improvement.

Due to space limitations, we report the results and experiment details of text detoxification under both automatic and human evaluation in Appendix C.1.

4.6 Human Evaluation

To better verify the effectiveness of KEST, we also conduct a human evaluation. For each model, we generated 100 samples on each task. We invite 6 competent annotators to score these samples on three criteria – **Fluency**, **Novelty**, and **Attribute Relevance** in a blind review manner. As shown in Table 2, KEST consistently outperforms the best baseline (UniLM-PT(select)+PL) on all three metrics, which indicates that KEST not only has better controllability over attributes but also generates fluent and diverse texts. See Appendix A.5 for detailed evaluation descriptions and metrics.

4.7 Ablation Study

We conduct an ablation study on the AGNews dataset and compare different KEST variants. As shown in Table 3, we can find: 1) Soft pseudo text prominently improves PPL and diversity, outperforming the hard one. As discussed in Sec. 3.3, such soft PT could bring smoother noise and help further push the learned distribution boundary. 2) Kernel-based learning alleviates the over-exploitation problem of the traditional cross-entropy loss and further enhances inner-group diversity (-0.93 S-BLEU), empirically supporting Theorem 1. 3) KEST's NAG ability not only reduces time complexity but also slightly boosts fluency (-0.75 O-PPL). However, the diversity improvement attributed to NAG is correlated to the noisy level. Only with an appropriate masking probability p_m could NAG facilitate more diverse text (see Fig. 4). Besides, pseudo text notably promotes all metrics, verifying our claim in Sec. 1 that such synthetic pseudo text leads to further improvement beyond pseudo labels.

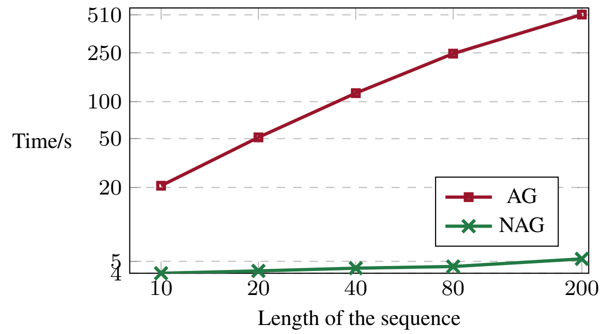

Figure 2: Comparison of decoding time of NAG and AG for 100 pseudo text batches (batch size=8) with different text lengths.

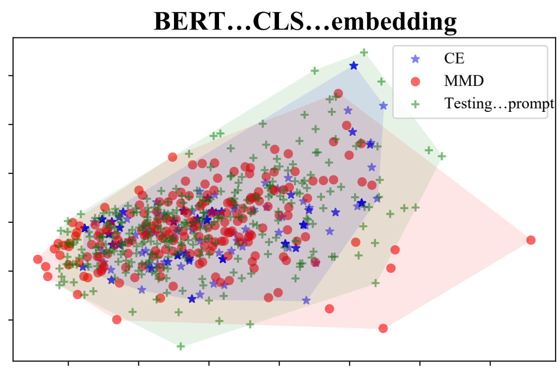

Figure 3: BERT *[CLS]* embedding of generated texts from KEST using cross-entropy (CE) and our MMD loss $\mathcal{L}_{ker}$ respectively.

4.8 Analysis

Time Consumption. Fig. 2 shows the decoding time of our NAG and AG generators for generating pseudo text with different text lengths. We found that the time costs of the AG module increase almost linearly w.r.t. the text length. In comparison, our NAG generator $\mathcal{G}_{nag}$ greatly accelerates the generation of pseudo text, especially when the sequence length is long. Furthermore, we compare the training time of KEST using $\mathcal{G}_{ag}$ and $\mathcal{G}_{nag}$, respectively. We observe that the latter achieves *1.2*× and *1.3*× speedup on IMDb and AGNews, respectively, which could be further improved with a larger ratio of pseudo text (e.g., the ratio could increase to *6.7*× with 1/10 unlabeled data), making self-training more practical.

Effect of Kernel-based Learning. To analyze the effect of our kernel distance loss $\mathcal{L}_{ker}$ in Eq. (5), we train two models for 5 epochs with only pseudo text given the same prompt and starting checkpoint using the kernel loss and the traditional cross-entropy loss, respectively. We then visualize the text

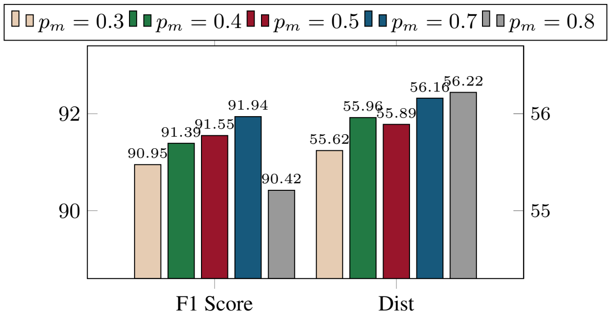

Figure 4: Results of KEST with different levels of mask ratio in AGNews Dataset.

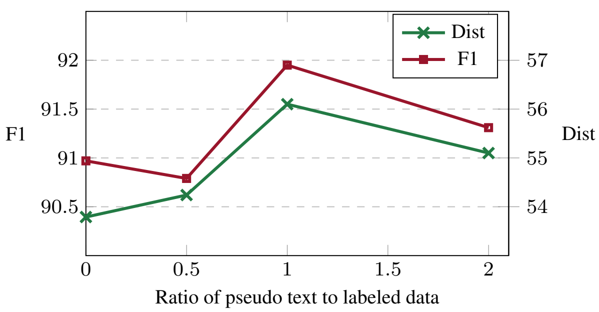

Figure 5: Generation controllability (F1) on a different number of pseudo text on AGNews dataset.

generated with given prompts from the two models by using corresponding BERT-large *[CLS]* embedding as text representations and plot them. As depicted in Fig. 3. We can find that the model trained with cross-entropy loss collapses in a smaller space than the training data space. In contrast, the one with kernel loss successfully extends the learned distribution, which helps explore a larger potential space towards the real one, corroborating our claim and theoretical analysis.

Effect of random mask ratio. The random mask ratio p_m is a hyperparameter that can control the noise level of generated pseudo text. Fig. 4 shows the generation performance of KEST with different mask ratios in the AGNews dataset. We find that a higher ratio leads to a more noisy and diverse generation. A moderately higher ratio also generally improves controllability. However, an extremely high p_m brings too much noise and hence obstructs learning. We achieve the best controllability with $p_m = 0.7$, indicating a suitable mask ratio is necessary to balance exploration and exploitation.

Number of pseudo text. We evaluate KEST on varying numbers of pseudo text, keeping all the other settings unchanged. As shown in Fig. 5, KEST performs the best with equal size of pseudo text and labeled data (Ratio = 1). More pseudo text brings more noise which hurts generation quality as the model captures more meaningless noise than semantics. Too little pseudo text makes the model lose exploration ability and thus fail to extend the learned distribution boundary, causing poor control accuracy and diversity. Therefore, a suitable ratio is essential to balance exploration and fluency.

Model	Generation
UniLM + PT (select) + PL	1) *Well, some people might* think that this film is a masterpiece. They are right. The film is not just a love story, but a love story. What I like about this film is that it shows a different side of women... 2) *Well, some people might* not like this film, but some people might. Well, most people would not like this movie. But the main reason I like it so much is that it has a lot of humor...
KEST	1)*Well, some people might* think it's a little over the top and the story is really predictable, but as I saw on TV in the early 90's I wasn't disappointed in this movie! While the plot is kind of predictable and the main character is supposed to be a guy, the whole thing has been made into a very cool and entertaining film... 2) *Well, some people might* think that this was a lot like "Jaws", or "Alien", or something like that. Sadly, it is not. I was lucky enough to see it. It's a very clever, intelligent and entertaining film with good performances...

Table 4: Samples generated with specified positive sentiment and input prompt '*Well, some people might*'. Words in blue/red are positive/negative indicators, respectively.

Case Study. In order to verify the generation quality and attribute relevance, we present some cases sampled from different models in Table 4. We can see that traditional ST methods (UniLM+PT(select)+PL) suffer from repeating phrases (e.g., "love story" and "not like"), exhibiting poor generation diversity and novelty. In contrast, KEST produces more diverse expressions thanks to kernel-based learning and smoother soft pseudo text while staying faithful to the given positive attribute. We present more generated cases on different tasks in Appendix E.

5 Conclusion

We propose a novel KEST method to incorporate Self-training into semi-supervised controllable NLG. KEST (1) applies a practical multi-task generator to generate soft pseudo text in parallel, significantly reducing decoding time while injecting soft noise to the text; (2) uses soft kernel-based loss to encourage exploration of the learned distribution and increase control accuracy and generation diversity. Theoretical analysis and empirical experiments demonstrate that KEST acts as a combination of regularization-like exploitation and attribute boundary exploration, improving control accuracy with satisfactory generation fluency, diversity, and accelerated training. In the future, we plan to try more advanced NAG methods to improve the generation quality of the pseudo text.

Ethical Statement

Topic/sentiment-controlled generated text may contain biased or offensive expressions. Besides, our model can produce more plausible texts that could be utilized to create fake news and disinformation. However, these generated texts can also be used as pseudo data in data augmentation for fact-checking and fake news detection and thus have the potential to improve current fact-checking models.

Acknowledgements

Feng's and Lakshmanan's research was supported in part by grants from NSERC (Canada) and UBC Data Science Institute.

References

[Bhat *et al.*, 2021] Meghana Moorthy Bhat, Alessandro Sordoni, and Subhabrata Mukherjee. Self-training with few-shot rationalization. In *EMNLP 2021*, pages 10702–10712, November 2021.

[Chapelle *et al.*, 2006] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20, 2006.

[Chen *et al.*, 2021] Yiming Chen, Yan Zhang, Chen Zhang, Grandee Lee, Ran Cheng, and Haizhou Li. Revisiting self-training for few-shot learning of language model. In *EMNLP 2021*, pages 9125–9135, November 2021.

[Chien and Lee, 2021] Chung-Ming Chien and Hung-yi Lee. Hierarchical prosody modeling for non-autoregressive speech synthesis. In *SLT 2021*, pages 446–453, 2021.

[Dathathri *et al.*, 2020] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *ICLR*, 2020.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL 2019*, pages 4171–4186, June 2019.

[Dong *et al.*, 2019] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *NeurIPS 2019*, December 2019.

[Du *et al.*, 2021] Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. Self-training improves pre-training for natural language understanding. In *NAACL 2021*, pages 5408–5418, Online, June 2021.

[Gehman *et al.*, 2020] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of EMNLP 2020*, pages 3356–3369, Online, November 2020.

[Grandvalet and Bengio, 2004] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, volume 17, 2004.

[Gretton *et al.*, 2012] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[Gu *et al.*, 2018] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. In *ICLR*, 2018.

[Gururangan *et al.*, 2020] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *ACL 2020*, pages 8342–8360, Online, July 2020.

[Han *et al.*, 2019] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *ICCV 2019*, pages 5138–5147, 2019.

[He *et al.*, 2020] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation. In *ICLR*, 2020.

[Jiao *et al.*, 2021] Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael Lyu, and Irwin King. Self-training sampling with monolingual data uncertainty for neural machine translation. In *ACL 2021*, pages 2840–2850, 2021.

[Kahn *et al.*, 2020] Jacob Kahn, Ann Lee, and Awni Hannun. Self-training for end-to-end speech recognition. In *ICASSP 2020*, pages 7084–7088. IEEE, 2020.

[Keskar *et al.*, 2019] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858, 2019.

[Krause *et al.*, 2021] Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. Gedi: Generative discriminator guided sequence generation. In *Findings of EMNLP 2021*, pages 4929–4952, 2021.

[Li *et al.*, 2016] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *NAACL 2016*, pages 110–119, June 2016.

[Li *et al.*, 2021] Shiyang Li, Semih Yavuz, Wenhu Chen, and Xifeng Yan. Task-adaptive pre-training and self-training are complementary for natural language understanding. In *Findings of EMNLP 2021*, pages 1006–1015, 2021.

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

[Liu *et al.*, 2021] Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *ACL 2021*, pages 6691–6706, Online, August 2021.

[Liu *et al.*, 2022] Puyuan Liu, Chenyang Huang, and Lili Mou. Learning non-autoregressive models from search for unsupervised sentence summarization. In *ACL 2022*, pages 7916–7929, 2022.

[Loshchilov and Hutter, 2019] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[Ma *et al.*, 2019] Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. Flowseq: Non-autoregressive conditional sequence generation with generative flow. In *EMNLP-IJCNLP*, pages 4282–4292, 2019.

[Maas *et al.*, 2011] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL 2011*, pages 142–150, 2011.

[Mukherjee and Hassan Awadallah, 2020] Subhabrata Mukherjee and Ahmed Hassan Awadallah. Uncertainty-aware self-training for few-shot text classification. In *NeurIPS 2020*, Online, 2020.

[Park *et al.*, 2020] Daniel S. Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V. Le. Improved noisy student training for automatic speech recognition. In *Interspeech 2020*. ISCA, Oct 2020.

[Qian *et al.*, 2022] Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. Controllable natural language generation with contrastive prefixes. In *Findings of ACL 2022*, pages 2912–2924, May 2022.

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

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020.

[Schick and Schütze, 2021] Timo Schick and Hinrich Schütze. Generating datasets with pretrained language models. In *EMNLP 2021*, pages 6943–6951, 2021.

[Scudder, 1965] Henry Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[Sohn *et al.*, 2020] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *NeurIPS*, 33:596–608, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[Vu *et al.*, 2021] Tu Vu, Minh-Thang Luong, Quoc Le, Grady Simon, and Mohit Iyyer. STraTA: Self-training with task augmentation for better few-shot learning. In *EMNLP 2021*, pages 5715–5731, November 2021.

[Wang *et al.*, 2020] Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. Adaptive self-training for few-shot neural sequence labeling. *arXiv preprint arXiv:2010.03680*, 2020.

[Wei *et al.*, 2021a] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *CVPR*, pages 10857–10866, 2021.

[Wei *et al.*, 2021b] Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. In *ICLR*, 2021.

[Xie *et al.*, 2020] Qizhe Xie, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *CVPR 2020*, pages 10684–10695, 2020.

[Yang *et al.*, 2020] Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. Generative data augmentation for commonsense reasoning. In *EMNLP Findings*, pages 1008–1025, 2020.

[Yang *et al.*, 2023] Zonghan Yang, Xiaoyuan Yi, Peng Li, Yang Liu, and Xing Xie. Unified detoxifying and debiasing in language generation via inference-time adaptive optimization. In *ICLR*, 2023.

[Yarowsky, 1995] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*, pages 189–196, 1995.

[Yogatama *et al.*, 2019] Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*, 2019.

[Zhang and Zong, 2016] Jiajun Zhang and Chengqing Zong. Exploiting source-side monolingual data in neural machine translation. In *EMNLP*, pages 1535–1545, 2016.

[Zhang *et al.*, 2015] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

[Zhang *et al.*, 2021] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. Revisiting few-sample BERT fine-tuning. In *ICLR*, 2021.

[Zhang *et al.*, 2022] Shuai Zhang, Meng Wang, Sijia Liu, Pin-Yu Chen, and Jinjun Xiong. How unlabeled data improve generalization in self-training? a one-hidden-layer theoretical analysis. In *ICLR*, 2022.

[Zhu *et al.*, 2018] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *SIGIR 2018*, page 1097–1100, 2018.