ZeroAE: Pre-trained Language Model based Autoencoder for Transductive Zero-shot Text Classification

Kaihao Guo[1,2*], **Hang Yu**[1*], **Cong Liao**[1], **Jianguo Li**[1†], **Haipeng Zhang**[2†]
[1]Ant Group, China
[2]School of Information Science and Technology, ShanghaiTech University, China
{guokh,zhanghp}@shanghaitech.edu.cn
{hyu.hugo, liaocong.lc, lijg.zero}@antgroup.com

Abstract

Many text classification tasks require handling unseen domains with plenty of unlabeled data, thus giving rise to the self-adaption or the so-called transductive zero-shot learning (TZSL) problem. However, current methods based solely on encoders or decoders overlook the possibility that these two modules may promote each other. As a first effort to bridge this gap, we propose an autoencoder named ZeroAE. Specifically, the text is encoded with two separate BERT-based encoders into two disentangled spaces, i.e., label-relevant (for classification) and label-irrelevant respectively. The two latent spaces are then decoded by prompting GPT-2 to recover the text as well as to further generate text with labels in the unseen domains to train the encoder in turn. To better exploit the unlabeled data, a novel indirect uncertainty-aware sampling (IUAS) approach is proposed to train ZeroAE. Extensive experiments show that ZeroAE largely surpasses the SOTA methods by 15.93% and 8.70% on average respectively in the label-partially-unseen and label-fully-unseen scenario. Notably, the label-fully-unseen ZeroAE even possesses superior performance to the label-partially-unseen SOTA methods.[1]

1 Introduction

Collecting human-labeled data often comes at a high cost for many NLP tasks, since it typically requires domain expertise and massive labeling efforts (Beltagy et al., 2022). It is therefore desirable and beneficial to consider the challenging (generalized) zero-shot learning (ZSL) that aims to adapt a learner to unseen domains or even unseen tasks without any annotated data (Zhang et al., 2020). In particular for the text classification problem, Yin et al. defines ZSL under two scenarios: label-partially-unseen and label-fully-unseen. The former demands domain adaptation that generalizes the model to classify text of unseen classes whose labeled data are unavailable during training. As a further step, the latter requires general-purpose ZSL models for new task adaption without requiring labeled data at all. The key to ZSL lies in improving the generalization performance by utilizing external knowledge (Chen et al., 2021).

Since pretrained language models (PLMs) memorize rich sources of external knowledge when being pretrained on a large text corpus, they can serve as an extremely powerful hammer for ZSL. Existing PLM-based ZSL approaches concentrate on either encoder-based (i.e., discriminative) or decoder-based (i.e., generative) models.[2] Specifically, encoder-based methods (Yin et al., 2019; Ye et al., 2020; Liu et al., 2021a; Alcoforado et al., 2022) typically treat text classification as a text entailment (TE) or a QA task, and fine-tune BERT or RoBERTa (Liu et al., 2019) as the embedding function to match the texts and labels in the seen classes. These methods then generalize to the unseen classes using the same embedding function and select labels that can best match the text semantically. As pointed out in (Ma et al., 2021), the BERT-based models could suffer from the issue of large uncertainty for unseen class generalization. Hence, labeled data are still required to stabilize the performance. On the other hand, decoder-based methods (Ye et al., 2022; Gao et al., 2022) attack the ZSL problem from the aspect of data augmentation. They employ GPT-2 to generate training data for the unseen classes given the labels, and next train a classifier based on the augmented data. Unfortunately, the data yielded by GPT-2 (i.e., the decoder-based methods) may contain a large portion of low-quality samples that are detrimental

[*] Equal contribution. This work was done when Kaihao Guo was a research intern at Ant Group.

[†] Corresponding author.

[1] Code is available at https://github.com/ant-research/plm_based_autoencoder_zero_shot_text_classification

[2] A more comprehensive review is provided in Appendix A.

Findings of the Association for Computational Linguistics: ACL 2023, pages 3202–3219
July 9-14, 2023 ©2023 Association for Computational Linguistics

to the training of the classifier. Worse still, GPT-2 in these approaches cannot be fine-tuned in an end-to-end manner to soften this issue. Although GPT-3 can alleviate the data quality problem to some extent, the dauntingly huge size precludes its widespread use (Brown et al., 2020). One promising solution to the above issues is to combine the encoder and the decoder-based models: the former may help to filter out the data of low quality given by the latter, while the latter can generate data with labels to boost the performance of the former.

Apart from PLMs, another source of external knowledge is unlabeled data. In practice, we often have access to abundant unlabeled data, and such data can assist in familiarizing the domain (or task) agnostic PLMs with the target domain (or task) (Rahman et al., 2019; Gera et al., 2022). The resulting zero-shot learning with unlabeled data is called transductive (generalized) zero-shot learning (TZSL). One appealing approach for TZSL is to involve the unlabeled data in a self-training loop (Ye et al., 2020; Wang et al., 2021a, 2022; Gera et al., 2022) by iterating between 1) estimating pseudo labels for all unlabeled data given an encoder-based model (e.g., a BERT-based TE model) and 2) refining the encoder-based model using the pseudo labels with high confidence. However, these self-training methods may lead to the problem of error accumulation (Wang and Breckon, 2020), that is, the mistakenly pseudo-labeled data in one iteration can severely affect subsequent iterations and the final predictions.

In this paper, we propose an autoencoder framework for TZSL. It harnesses the strength of both PLMs and unlabeled data, while at the same time bringing the best from the encoder-based and the decoder-based methods together. We name the resulting model ZeroAE and it is to our knowledge the first approach that aims to solve the TZSL problem in NLP from the perspective of autoencoders. Particularly, we specify the encoder and the decoder to be fine-tuned BERT and GPT-2 respectively. To enable the two PLMs to promote each other and to further self-adapt to the task at hand, we design two main types of data flows in ZeroAE: text reconstruction flow and label reconstruction flow. The first one aims to recover the text data after inputting it to the encoder and subsequently the decoder, while the second tries to recover the label after first generating text given the label via the decoder and then predicting the label given the generated text via the encoder. Furthermore, we assume that the latent space can be split into two parts: label-relevant and label-irrelevant. These two parts are discrete (vector-quantized), disentangled, and are given by two different encoders (i.e., fine-tuned BERT). Only the label-relevant part is used for classification to remove the interference from the label-irrelevant part, while both parts are required for text reconstruction and generation. Additionally, to better handle the unlabeled data, we also adopt contrastive learning, and further propose a simple yet effective method named indirect uncertainty-aware sampling (IUAS) to train ZeroAE, allowing the model to pay more attention to those unlabeled data with high uncertainty as the training process proceeds and lowering down the uncertainty with the assistance of GPT-2.

In summary, our key contributions are:

- To our best knowledge, we are among the first to propose an end-to-end autoencoder, ZeroAE, for TZSL in NLP, which seamlessly integrates the encoder and decoder-based models. By designing the text and label reconstruction flows, we allow BERT and GPT-2 to promote each other and equip them with the capability of auto-calibration to unseen domains and tasks.
- We propose a novel method named IUAS to train ZeroAE, gradually focusing on those unlabeled data with high uncertainty and reducing the uncertainty with the help of GPT-2.
- We further incorporate several advanced techniques into ZeroAE to boost its performance, including contrastive learning, latent space discretization, disentanglement, and prompting.
- We demonstrate the usefulness of ZeroAE through extensive experiments on four real-world datasets for TZSL under the two settings: label-partially-unseen and label-fully-unseen. ZeroAE greatly outperforms the existing PLM-based methods by 15.93% and 8.70% on average respectively in the two settings. Remarkably enough, ZeroAE without labels is even superior to the existing methods with labels.

2 ZeroAE

We begin this section by defining the transductive (generalized) zero-shot learning (TZSL) problem. Let $\mathbb{D}^S = \{(\boldsymbol{x}_i^S, \boldsymbol{y}_i^S)\}$ and $\mathbb{D}^U = \{(\boldsymbol{x}_i^U, \boldsymbol{y}_i^U)\}$ denote the data for seen and unseen classes respectively, where $\boldsymbol{x}_i$ is the text, $\boldsymbol{y}_i$ is the label corre-

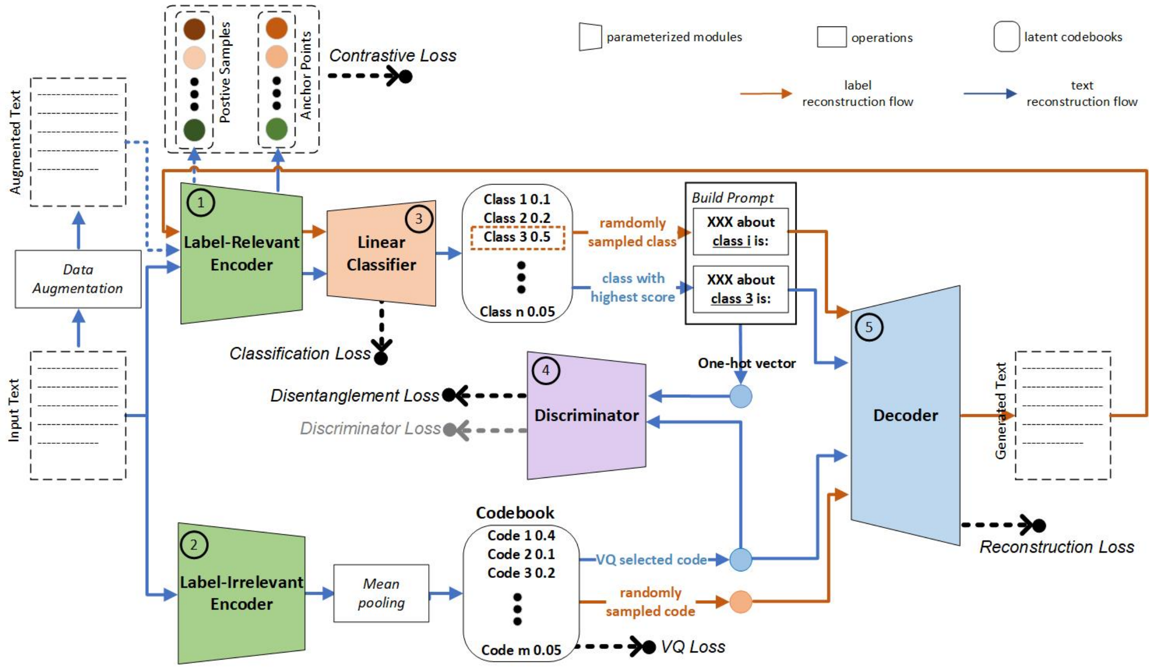


Figure 1: The overall architecture of ZeroAE. ZeroAE consists of five modules, including ① a label-relevant encoder, ② a label-irrelevant encoder, ③ a classifier, ④ a discriminator, and ⑤ a decoder.

sponding to $\boldsymbol{x}_i$, and i is the sample index. Note that $\boldsymbol{y}_i$ can only take values from $\{\mathrm{y}_1, \cdots, \mathrm{y}_C\}$, where y_j denotes the class name for class j and C is the number of classes. For *label-partially-unseen* TZSL, labeled data from $\mathbb{D}^S$ and unlabeled data from $\mathbb{D}^U$ are utilized to train the model. We then test the model under the assumption that the labels are from both seen and unseen classes. The objective of label-partially-unseen TZSL is to conduct domain adaption automatically from seen to unseen classes. On the other hand, for *label-fully-unseen* TZSL (a.k.a. extremely weakly supervised learning (Wang et al., 2022)), we take one step further, and only use unlabeled data from both $\mathbb{D}^S$ and $\mathbb{D}^U$ to train the model. This challenging task requires the model to self-adapt to the new task at hand given no labeling information for the task.

In order to solve the TZSL problem, we propose ZeroAE in this paper, whose overall architecture is shown in Figure 1. As mentioned in the introduction, there are two major types of data flows in ZeroAE, namely, the text reconstruction flow (denoted by the blue arrows) and the label reconstruction flow (denoted by the orange arrows). To provide an illuminating overview of ZeroAE, we mainly follow the text reconstruction flow and describe an example of how ZeroAE processes one sentence $\boldsymbol{x}_i$ for the purpose of topic categorization. Suppose that the text $\boldsymbol{x}_i$ is "Could stress or an unhealthy diet trigger lung cancer?". We first encode the text into two disentangled latent spaces $\boldsymbol{z}^R$ and $\boldsymbol{z}^I$ with two separate encoders (see ① and ②). The first latent space $\boldsymbol{z}^R$ characterizes the label-relevant information, such as "stress", "unhealthy diet", and "lung cancer", while the second $\boldsymbol{z}^I$ represents the label-irrelevant information, such as the syntax "Could ... ?". The label-relevant information is further inputted into a classifier (see ③) in order to find the correct topic, i.e., "health". A discriminator (see ④) is introduced to guarantee that the two latent spaces are disentangled so the label-irrelevant information cannot adversely affect the classifier. Finally, the two latent embeddings are fed into the decoder (see ⑤) so as to recover the original text $\boldsymbol{x}_i$. Next, we elaborate on each module of ZeroAE. For ease of exposition, all notations in this paper are summarized in Table 10.

2.1 Label-Relevant Encoder and Classifier

The label-relevant encoder Enc^R is a fine-tuned BERT (L=12, H=768, total parameters=110M). We fix the first 10 layers and fine-tune the remaining two layers. Concretely, we follow the framework of text entailment (Yin et al., 2019), and pack the input text $\boldsymbol{x}_i$ with all C candidate labels $\mathrm{y}_{1:C}$ like "[CLS] $\boldsymbol{x}_i$ [SEP] hypothesis of y_j [SEP]", where $j = 1, \cdots, C$. As Enc^R aims to extract features for classification, we regard the [CLS] token in

BERT as the output of the encoder, namely,

$$\boldsymbol{z}_{ij}^{R} = \mathrm{Enc}^{R}(\boldsymbol{x}_i, \mathrm{y}_j). \tag{1}$$

For labeled and generated data, we further input the embeddings from all candidate classes $[\boldsymbol{z}_{i1}^{R}, ..., \boldsymbol{z}_{iC}^{R}]$ to a linear classifier Cls, and obtain the C-dimensional vector $\boldsymbol{s}_i^R$ of the classification probability as:

$$\boldsymbol{s}_i^R = \mathrm{Cls}([\boldsymbol{z}_{i1}^{R}, ..., \boldsymbol{z}_{iC}^{R}]). \tag{2}$$

The corresponding *classification loss* is:

$$\mathcal{L}_{cls} = \mathcal{H}(f(\boldsymbol{y}_i), \boldsymbol{s}_i^R), \tag{3}$$

where $\mathcal{H}$ represents the cross entropy, and $f(\cdot)$ is a function that converts $\boldsymbol{y}_i$ to a one-hot vector whose j-th element equals 1 if $\boldsymbol{y}_i = \mathrm{y}_j$. Meanwhile, the predicted class $\hat{\boldsymbol{y}}_i$ for $\boldsymbol{x}_i$ is given by $\hat{\boldsymbol{y}}_i = \mathrm{y}_c$, where the index $c = \arg\max_j s_{ij}^R$ and s_{ij}^R denotes element j in the vector $\boldsymbol{s}_i^R$.

Additionally, contrastive learning is also applied to train Enc^R in order to enhance the performance of ZeroAE on the unlabeled data. Given the text data $\boldsymbol{x}_i$, the easy data augmentation (EDA) method (Wei and Zou, 2019) is exploited to generate the positive (i.e., similar) samples $\boldsymbol{x}_i'$. Specifically, we perform three random operations on $\boldsymbol{x}_i$ with probability 0.1, including synonym replacement, random insertion, and random deletion of words. We do not use the swap-two-words operation in (Wei and Zou, 2019) though, so as to retain the semantic structure of the sentences. On the other hand, the negative (i.e., disimilar) samples of $\boldsymbol{x}_i$ are chosen as the remaining texts $\boldsymbol{x}_j$ in the same batch. The resulting *contrastive loss* can be expressed as:

$$\mathcal{L}_{con} = \frac{\cos\left(\boldsymbol{z}_{ic}^{R}, \boldsymbol{z}_{ic}^{R\prime}\right)}{\sum_{j\neq i}\cos\left(\boldsymbol{z}_{ic}^{R}, \boldsymbol{z}_{jc}^{R}\right)}, \tag{4}$$

where $\boldsymbol{z}_{ic}^{R} = \mathrm{Enc}^R(\boldsymbol{x}_i, \mathrm{y}_c)$, $\hat{\boldsymbol{y}}_i = \mathrm{y}_c$ is the predicted class for $\boldsymbol{x}_i$, and cos denotes cosine similarity. Minimizing the above loss yields an embedding space where the semantically similar text pairs are nearby whereas the dissimilar ones are distant from each other, and offers the opportunity for discovering the decision boundaries between different classes.

2.2 Label-Irrelevant Encoder

The label-irrelevant encoder Enc^I is also a fine-tuned BERT. As the task here is to extract label-irrelevant features, which differs substantially from the pretraining tasks of BERT, we fine-tune the total 12 layers. Since the label-irrelevant features such as the syntax are related to the entire sentence, we use the mean pooling of the last layer as the embedding of the text. In addition, similar to the output $\boldsymbol{z}^R$ of Enc^R, we also discretize this latent space $\boldsymbol{z}^I$, by means of vector quantization (VQ) (Van Den Oord et al., 2017). Discrete latent space is proven to be advantageous to its continuous counterpart for text generation due to the discrete nature of NLP (Ji and Huang, 2021). VQ helps the latent space model to circumvent the issue of posterior collapse (i.e., the latent variables are ignored in the decoder), which often plagues pretrained VAEs (Li et al., 2020; Xu et al., 2020). Concretely, we introduce a codebook $\boldsymbol{e} = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_K]$ with size $K = 32$ to represent the discrete latent space as shown in Figure 1. The output of the encoder $\mathrm{Enc}^I(\boldsymbol{x}_i)$ is compared to the codebook, and the codeword $\boldsymbol{e}_k$ closest to $\mathrm{Enc}^I(\boldsymbol{x}_i)$ in terms of Euclidean distance is chosen as the latent representation of $\boldsymbol{x}_i$. Framed mathematically,

$$\boldsymbol{z}_i^I = \arg\min_{\boldsymbol{e}_k\in\boldsymbol{e}} \| \mathrm{Enc}^I(\boldsymbol{x}_i) - \boldsymbol{e}_k\|_2^2. \tag{5}$$

The codebook is updated along with the parameters of Enc^I, in analogy to k-means clustering, to minimize the within-cluster distance. The corresponding *VQ loss* can be written as:

$$\begin{aligned}\mathcal{L}_{vq} =& \| \mathrm{sg}[\mathrm{Enc}^I(\boldsymbol{x}_i)] - \boldsymbol{e}\|_2^2 \\ &+ \beta\| \mathrm{Enc}^I(\boldsymbol{x}_i) - \mathrm{sg}[\boldsymbol{e}]\|_2^2,\end{aligned} \tag{6}$$

where sg stands for the operation “stop gradient” that prevents the gradient from flowing through that part of the equation. The first term fixes the encoder and aligns the codebook $\boldsymbol{e}$ such that the K codewords inside are as close to the encoder output $\mathrm{sg}[\mathrm{Enc}^I(\boldsymbol{x}_i)]$ as possible. The second term in turn fixes the codebook and updates the parameters of the encoder such that the encoder output commits as much as possible to its closest codeword. β here is a tuning parameter dictating the importance of the second term. We follow (Van Den Oord et al., 2017) to set $\beta = 0.25$.

2.3 Discriminator for Latent Space Disentanglement

To mitigate the possible negative impact of $\boldsymbol{z}^I$ on the classifier, we borrow the idea from factor VAE (Kim and Mnih, 2018) and encourage the two latent spaces $\boldsymbol{z}^R$ and $\boldsymbol{z}^I$ to be disentangled, that is, $q(\boldsymbol{z}^R, \boldsymbol{z}^I) = q(\boldsymbol{z}^R)q(\boldsymbol{z}^I)$.[3] Let a positive sample $\boldsymbol{z}^+$ denote the concatenated label-relevant and label-irrelevant embeddings $[f(\hat{\boldsymbol{y}}_i), \boldsymbol{z}_i^I]$ resulting from the same text $\boldsymbol{x}_i$ and a negative sample $\boldsymbol{z}^-$ denote the concatenated embeddings $[f(\hat{\boldsymbol{y}}_i), \boldsymbol{z}_j^I]$ that are never from the same text in one epoch.[4] The independence between $\boldsymbol{z}^R$ and $\boldsymbol{z}^I$ can be achieved by firstly training a discriminator to distinguish the positive samples from the negative ones and secondly training the remaining parts of ZeroAE to fool the discriminator. These two steps are iterated in every epoch. To this end, the *discriminator loss* function for training the discriminator can be expressed as:

$$\mathcal{L}_{disc} = -\log\Big(\mathrm{Disc}(\boldsymbol{z}^+)\big(1 - \mathrm{Disc}(\boldsymbol{z}^-)\big)\Big), \quad (7)$$

where Disc stands for the linear discriminator. The *disentanglement loss* for training the remaining parts of ZeroAE can be written as:

$$\mathcal{L}_{dise} = -\log\big(\mathrm{Disc}(\boldsymbol{z}^-)\big). \quad (8)$$

Note that we randomly sample M pairs of $(\boldsymbol{z}^+, \boldsymbol{z}^-)$ in each epoch and average the above two losses over them.

2.4 Decoder

After obtaining the label-relevant and label-irrelevant embeddings $(\boldsymbol{z}_i^R, \boldsymbol{z}_i^I)$ of text $\boldsymbol{x}_i$, we aim to reconstruct the text given the two latent embeddings by means of the decoder, that is,

$$\hat{\boldsymbol{x}}_i = \mathrm{Dec}(g(\hat{\boldsymbol{y}}_i), \boldsymbol{z}_i^I), \quad (9)$$

where Dec is a GPT-2 (L=12, H=768, total parameters=124M). We fix the first six layers and fine-tune the other six in GPT-2 when training ZeroAE.

In addition, the first input to the decoder is $g(\hat{\boldsymbol{y}}_i)$, where the predicted label $\hat{\boldsymbol{y}}_i$ is derived from $\boldsymbol{z}_i^R$ (cf. Section 2.1), and $g(\cdot)$ represents the prompting function that projects the label name to a sentence via a template. In the example of topic categorization, suppose the topic $\hat{\boldsymbol{y}}_i$ is "health" and the template is "The news with topic $\hat{\boldsymbol{y}}_i$ is: ", then the prompting function g will return the sentence "The news with topic health is: ". Note that we use the true and the predicted label name (i.e., $\boldsymbol{y}_i$ and $\hat{\boldsymbol{y}}_i$) respectively for the labeled and unlabeled data as the input to g. The sentence given by g then guides GPT-2 to generate $\hat{\boldsymbol{x}}_i$ by acting as the initial condition in the autoregressive model. The merit of using $g(\hat{\boldsymbol{y}}_i)$ is that it converts the "black box" latent variable $\boldsymbol{z}_i^R$ to a sentence that can be directly interpreted by GPT-2, even without any fine-tuning, greatly facilitating the reconstruction and generation of text related to the label.

On the other hand, the label-irrelevant latent variable $\boldsymbol{z}_i^I$ is fed into GPT-2 via cross attention and serves as the key and value in the attention mechanism. Hence, GPT-2 can generate sentences in light of the label-irrelevant features. Once we obtain the reconstructed text $\hat{\boldsymbol{x}}_i$ from Eq. (9), we can compute the *reconstruction loss* as the cross entropy between the original and the reconstructed text:

$$\mathcal{L}_{rec} = \sum\nolimits_j \mathcal{H}(f(x_{ij}), \hat{x}_{ij}), \quad (10)$$

where we abuse the notations $\hat{x}_{ij}$ for convenience to denote the estimated probability of word j in the reconstructed text $\hat{\boldsymbol{x}}_i$ taking the values in a predefined vocabulary, and $f(x_{ij})$ represents the one-hot vector corresponding to the true word.

In a nutshell, the overall objective function can be written as:

$$\min_{\Theta} \mathcal{L}_{disc}(\Theta) + \min_{\Psi} \mathcal{L}_{rem}(\Theta, \Psi), \quad (11)$$

where

$$\mathcal{L}_{rem} = \mathcal{L}_{rec} + \mathcal{L}_{vq} + \mathcal{L}_{con} + \mathcal{L}_{dise} + \mathcal{L}_{cls}, \quad (12)$$

and Θ and Ψ respectively denote the parameters of the discriminator and the remaining parts of ZeroAE. Note that in the above expression (11) we recursively update the discriminator (parameterized by Θ) and the remaining parts of ZeroAE (parameterized by Ψ), in a similar manner to GAN.

2.5 Indirect Uncertainty-Aware Sampling

It follows from the above discussion that the labeled data are invoked in all loss functions in Eq. (12), and the unlabeled data are concerned with all but the classification loss, while the generated data are only used in the classification loss. Note that both the labeled and generated data are associated with labels. Unlabeled data, nevertheless, may present high classification bias and uncertainty during training. Pseudo labeling (Wang

[3] We use q here following the notations in VAE. $q(\boldsymbol{z}) = \int p(\boldsymbol{x})q(\boldsymbol{z}|\boldsymbol{x})d\boldsymbol{x}$, where $p(\boldsymbol{x})$ denotes the distribution of the observed text and $q(\boldsymbol{z}|\boldsymbol{x})$ denotes the outputs of the encoders.

[4] To increase the number of positive and negative samples and to stabilize the performance of the discriminator, we further define $\boldsymbol{z}^+ = [f(\hat{\boldsymbol{y}}_i), \boldsymbol{z}_{i1}^I, \boldsymbol{z}_{i2}^I]$ and $\boldsymbol{z}^- = [f(\hat{\boldsymbol{y}}_i), \boldsymbol{z}_{i1}^I, \boldsymbol{z}_j^I]$, where $\boldsymbol{z}_{i1}^I$ and $\boldsymbol{z}_{i2}^I$ are randomly chosen from those that can be paired with $\boldsymbol{z}_i^R$ in this epoch, and $\boldsymbol{z}_j^I$ is randomly chosen from those that can never be paired with $\boldsymbol{z}_i^R$.

et al., 2021a) may be helpful to reduce the uncertainty, but it typically leads to the problem of error accumulation (i.e., bias) (Wang and Breckon, 2020). Different from pseudo labeling, we borrow the ideas from curriculum learning (Soviany et al., 2022) and uncertainty sampling in active learning (Aguilar et al., 2021) and propose an indirect uncertainty-aware sampling (IUAS) procedure to train ZeroAE. Viewed one way from curriculum learning, we intend to concentrate more on the "hard" samples with high uncertainty as the training process proceeds. Curriculum learning is known to achieve higher convergence speed and better accuracy without extra computational cost (Soviany et al., 2022). Viewed another way from uncertainty sampling, in order to reduce the uncertainty, we would like to simulate similar samples from the decoder GPT-2. These generated data have labels, which can help the encoder to better distinguish the unlabeled data with high uncertainty.

To move forward to these goals, we propose to conduct data selection by removing the unlabeled data whose probability of belonging to a class is larger than a threshold $\tau = 0.8$ at the beginning of every training epoch. In other words, we remove the data with small uncertainty but retain those with large uncertainty. Owing to the text reconstruction flow, GPT-2 can be gradually fine-tuned to generate samples similar to the retained unlabeled data. Meanwhile, in the label reconstruction flow, we further store the generated data from all previous epochs (denoted as $\mathbb{D}^G$), randomly pick CK samples in each epoch from $\mathbb{D}^G$, and use them to train the label-relevant encoder and the classifier, where C is the number of classes and K is the size of the VQ codebook. It is noteworthy that GPT-2 with the prompt acts as regularization here: the data generated by GPT-2 are usually associated with proper labels since half of the layers in GPT-2 are fixed, and using the generated data to train the classifier mitigates the problem of error accumulation in pseudo-labeling. Indeed, as shown in our experiments, IUAS outperforms pseudo labeling when coupled with ZeroAE. The overall training procedure is summarized in Algorithm 1 in the appendix.

3 Experiments

3.1 Datasets and Experiment Setup

We investigate the effectiveness of ZeroAE on four real-world datsets, including "Topic", "Situation", "Emotion", and "Complaint", under both label-partially-unseen and label-fully-unseen scenarios. The first three datasets are often used for benchmarking different zero-shot text classification approaches (Yin et al., 2019), and the last one aims to assign the customer complaints regarding Alipay reported by users to the corresponding response teams. Note that in the label-partially-unseen case, two different versions (v0 and v1) of the first three datasets are provided in (Yin et al., 2019) with non-overlapping labels, in order to prevent the models from overfitting to some classes. The detailed statistics regarding how the datasets are split into training, validation, and testing sets are summarized in Table 1. Please refer to Appendix B for more details on the datasets.

Table 1: Dataset statistics.

Dataset	Version	Seen classes		Unseen classes
		Train	Valid	Test
Topic	v0	650000	5000	50000
	v1	650000	5000	50000
Situation	v0	2428	240	689
	v1	1747	173	1102
Emotion	v0	20465	2405	5101
	v1	14204	1419	8901
Complaint	-	218	174	94

Experimental Setup: For the first three datasets, we use the pretrained BERT[5] and GPT-2[6] as the encoder and decoder in ZeroAE. For the fourth dataset of customer complaints, we use the pretrained ERNIE[7] and Chinese GPT-2[8], since texts in this dataset are in Chinese. The templates of the prompting function for the four datasets are also different, which is listed in Table 9 in the appendix. For optimization, we use Adam with learning rate 5×10^{-5} and the linear warm-up scheduler. To avoid overfitting, we resort to early stopping with a maximum number of 40 epochs. All results for ZeroAE shown below are averaged over five trials.

3.2 Results and Analysis

3.2.1 Label-partially-unseen TZSL

We first conduct experiments under the scenario of label-partially-unseen TZSL (Ye et al., 2020), as introduced at the beginning of Section 2. We juxtapose ZeroAE with four SOTA methods. The first three methods are encoder-based methods, and the third one further uses self-training to cope with

[5] https://huggingface.co/BERT-base-uncased

[6] https://huggingface.co/gpt2

[7] https://huggingface.co/nghuyong/ernie-1.0-base-zh

[8] https://github.com/Hansen06/GPT2-Chinese

Table 2: Macro F_1-score resulting from all benchmark methods for label-partially-unseen TZSL. ZeroAE achieves an improvement of 15.93% averaged over datasets and methods.

Methods	Topic		Situation		Emotion		Complaint
	v0	v1	v0	v1	v0	v1	
BERT	57.07	45.50	60.23	34.15	16.86	10.21	7.14
BERT-MNLI	54.37	45.80	63.74	50.13	30.21	21.40	-
BERT+RL	73.41	65.53	73.14	52.44	36.98	19.38	31.45
ZeroGen	64.71	54.34	67.97	52.67	26.38	22.44	26.07
ZeroAE-LPU	**75.32**	**71.75**	**78.58**	**71.54**	**42.71**	**30.75**	**37.19**
ZeroAE-LFU	69.68	66.11	72.87	63.36	31.25	23.31	20.49

unlabeled data, while the last one is a decoder-based method. More details are provided below:

1. **BERT** (Devlin et al., 2018): This approach directly uses BERT as a matching model without any fine-tuning.
2. **BERT-MNLI** (Yin et al., 2019): BERT is first pretrained on the MNLI dataset (Williams et al., 2018) and then fine-tuned on the training data with labels. Note that we cannot apply this method to the dataset of complaints as the customer complaints are in Chinese but BERT-MNLI is pretrained on texts in English.
3. **BERT+RL** (Ye et al., 2020): BERT plays the role of a pseudo labeler and reinforcement learning is utilized to select pseudo-labeled data automatically and further use them to refine BERT. Here the labeled data are used to fine-tune BERT and learn the data selection policy.
4. **ZeroGen** (Ye et al., 2022): GPT-2 is employed to generate samples for each unseen class. The generated data are then combined with the originally observed labeled data to train a classifier.

In addition, we consider two different settings of ZeroAE: the first one uses labeled data in the seen classes $\mathbb{D}^S$ in the same manner as in the above four methods, and the second only uses texts $\boldsymbol{x}_i$ in both $\mathbb{D}^S$ and $\mathbb{D}^U$ without any labels. We refer to the two settings as ZeroAE-LPU (label-partially-unseen) and ZeroAE-LFU (label-fully-unseen) respectively. We follow the SOTA methods to use macro F_1-score as the criterion to evaluate the performance[9], and the results are summarized in Table 2.

Remarkably, the proposed ZeroAE-LPU significantly outperforms the SOTA methods, achieving at least 1.91%, 5.44%, 5.73%, and 5.74% of gains in terms of the macro F_1-score respectively for the four datasets. The largest macro F_1-score increase can be as high as 19.10%. The second best approach, ZeroAE-LFU, also manifests a supremacy over the SOTA methods, even without using any information of labels at all. This bolsters our belief that it is quite beneficial to combine the encoder-based and decoder-based methods in a unified framework like ZeroAE.

As opposed to ZeroAE, the raw BERT model yields the worst results for all datasets, since the pretrained BERT without fine tuning cannot self-adapt to different tasks in practice. After fine-tuning with the labeled data for each dataset, BERT-MNLI greatly increases the macro F_1-score, but still compares unfavorably with ZeroAE, probably because it cannot well generalize to unseen classes (Ma et al., 2021). On the other hand, ZeroGen is on par with BERT-MNLI, showing the advantages of decoder-based models. The key caveat with ZeroGen though is that it may generate data with low quality, since GPT-2 is not fine-tuned to cope with the task. Hence, its macro F_1-score is worse than that of ZeroAE. Finally, it can be observed that BERT+RL typically performs better than BERT-MNLI and ZeroGen, after reaping benefits from the unlabeled data. However, this approach suffers from the problem of error accumulation as pointed out in the introduction. As a consequence, its performance deteriorates when it becomes difficult to clear-cut the decision boundaries between different classes semantically and the amount of labeled data are too small to provide sufficient supervision. This explains its deficiency in comparison with BERT-MNLI and ZeroGen for Emotion-v1 (see Appendix B for more details on this dataset). Note that differentiating between emotions semantically is a difficult task and the number of samples for the seen classes is relatively small in this dataset.

3.2.2 Label-fully-unseen TZSL

Next, we investigate the performance of ZeroAE when compared with other label-fully-unseen TZSL methods based on PLMs, including two

[9] The code of BERT+RL is not publicly accessible. To make a fair comparison, here we follow the experiment configuration and evaluation criterion in BERT+RL.

Table 3: Macro F_1-score for the ablation study in the case of label-partially-unseen TZSL. IUAS, and the classification, contrastive, and disentanglement loss are respectively removed from ZeroAE. The relative difference between the ablated and the original ZeroAE averaged over all datasets is shown in the last column. The results are averaged over 5 trials, and the standard deviation is presented in the brackets.

Methods	Topic		Situation		Emotion		Complaint	Average difference
	v0	v1	v0	v1	v0	v1		
ZeroAE (ours)	**75.32** (1.69)	**71.75** (1.00)	**78.58** (1.43)	71.54 (1.26)	**42.71** (2.35)	**30.75** (1.94)	**37.19** (1.14)	-
−IUAS	55.80 (6.63)	59.98 (1.67)	77.10 (1.46)	62.75 (2.54)	28.72 (4.28)	23.84 (2.45)	25.73 (1.48)	-10.56
−IUAS+Pseudo labeling	71.29 (0.91)	63.71 (1.79)	66.90 (3.71)	62.47 (3.16)	38.39 (2.01)	23.04 (0.86)	29.93 (3.41)	-7.44
−Classification Loss	69.68 (1.58)	66.11 (1.15)	72.87 (2.14)	63.36 (2.75)	31.25 (3.61)	23.31 (1.37)	20.49 (1.60)	-8.68
−Contrastive Loss	72.86 (1.06)	66.87 (1.95)	72.32 (3.19)	63.29 (4.52)	33.46 (2.02)	24.65 (0.72)	30.18 (1.91)	-6.31
−Disentanglement Loss	74.59 (1.47)	69.43 (1.38)	74.41 (1.56)	**73.12** (1.89)	36.48 (4.48)	29.78 (2.89)	36.27 (0.92)	-1.97

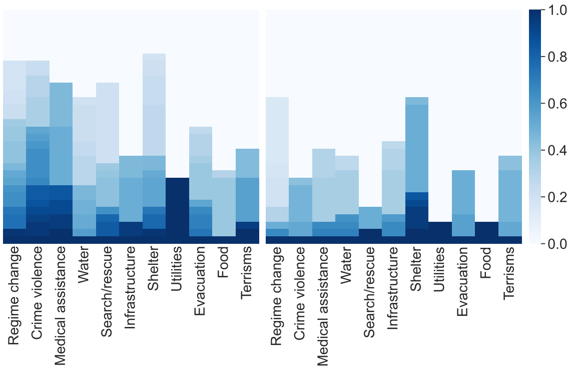
Table 4: Weighted F_1-score resulting from all benchmark methods for label-fully-unseen TZSL. ZeroAE achieves an improvement of 8.70% averaged over datasets and methods.

Methods	Topic	Situation	Emotion	Complaint
BERT-TE	45.70	45.23	25.20	5.79
P-ZSC	50.68	58.84	30.22	8.14
ZeroGen	60.15	**62.11**	24.25	3.14
ZeroAE	**62.96**	61.68	**32.41**	**12.53**

encoder-based methods and the decoder-based method ZeroGen. In this setting, we merge the training and testing data in both v0 and v1 for the datasets Topic, Situation, and Emotion. A summary of the benchmark methods is given below:

1. **BERT-TE** (Yin et al., 2019): This approach exploits pretrained BERT and formulates the TZSL problem as a text entailment task.
2. **P-ZSC** (Wang et al., 2022): The label names for all classes are first expanded by finding the most semantically similar words or phrases to them from a text corpus. Self-training is then applied by pseudo-labeling the data using a BERT-based matching algorithm that evaluates the similarity between the texts and the expanded label names.
3. **ZeroGen** (Ye et al., 2022): Different from the settings in the previous subsection where both labeled and generated data are used to train the classifier, here only generated data resulting from GPT-2 are used.

We follow the second and the third method to use weighted F_1-score as the evaluation criterion[10]. The results are shown in Table 4. Once again, ZeroAE markedly improves the weighted F_1-score by 8.70% on average. It achieves the best weighted F_1-score among all methods for three datasets and only slightly worse performance than ZeroGen for one dataset, suggesting that ZeroAE can well play the role of a general-purpose zero-shot learner that allows for auto-calibration to different tasks with the assistance of unlabeled data.



(a) +Discriminator (b) −Discriminator

Figure 2: Sorted selection probability for all VQ codewords associated with each class (a) with and (b) without the discriminator. Without the discriminator, each class is highly correlated with a few codewords, limiting the modeling power of ZeroAE.

3.2.3 Ablation Study

Impact of different modules in ZeroAE: We conduct an ablation study to verify the effectiveness of different modules in ZeroAE, and display the results in Table 3. More details regarding the experiment settings and results can be found in Appendix C. There are three major findings that can be gleaned from Table 3:

1. The training procedure IUAS contributes the most to the superior performance of ZeroAE on TZSL. Ablating IUAS from ZeroAE leads to a dramatic drop of 10.56% in terms of the macro F_1-score averaged over all datasets. Furthermore, by replacing IUAS with pseudo labeling, the resulting macro F_1-score is reduced by 7.44%. This observation implies that pseudo labeling can help ZeroAE to handle the unlabeled data, but IUAS is a better option, since it adopts GPT-2 as regularization and alleviates the issue of error accumulation in pseudo labeling.

[10] The corpus for label expansion in P-ZSC is not publicly accessible. For a fair comparison, we use the same experiment configuration and evaluation criterion as in P-ZSC.

Table 5: Ablation study on the PLM backbone.

Backbones	Topic		Situation		Emotion	
	v0	v1	v0	v1	v0	v1
BERT+GPT2	75.32	71.75	78.58	71.54	42.71	30.75
BART	76.49	74.11	77.14	69.51	45.37	28.47

Table 6: Ablation study on the number of fixed layers in the PLMs, including ① the label-relevant encoder, ② the label-irrelevant encoder, and ⑤ the decoder.

	#Fixed	Topic		Situation		Emotion	
		v0	v1	v0	v1	v0	v1
Module ①	0	71.23	67.54	78.47	70.84	41.20	24.39
	2	72.41	68.62	**79.21**	69.02	42.11	27.93
	6	73.95	70.11	78.38	**71.79**	42.52	29.19
	10	**75.32**	**71.75**	78.58	71.54	**42.71**	**30.75**
Module ②	0	75.32	71.75	**78.58**	71.54	**42.71**	**30.75**
	2	76.53	68.42	77.92	**72.54**	41.47	28.14
	6	74.39	70.64	75.24	70.15	35.78	18.55
	12	**75.41**	**72.98**	73.11	70.92	30.07	15.49
Module ⑤	0	74.23	67.81	77.61	68.86	40.77	**32.56**
	2	**75.55**	68.72	78.55	69.22	39.16	30.14
	6	75.32	**71.75**	**78.58**	**71.54**	**42.71**	30.75
	10	73.91	70.89	77.19	67.11	38.21	30.53

Table 7: Impact of the IUAS threshold τ.

Threshold	Topic		Situation		Emotion	
	v0	v1	v0	v1	v0	v1
0.75	**71.56**	63.44	**70.41**	60.50	36.77	17.88
0.8	71.29	**63.71**	66.90	**62.47**	38.39	**23.04**
0.85	68.79	63.89	67.89	61.34	**38.85**	13.41
0.9	69.01	61.18	68.12	60.82	34.53	20.59

Table 8: Impact of the codebook size.

Codebook size	Topic		Situation		Emotion	
	v0	v1	v0	v1	v0	v1
16	72.14	65.90	69.51	57.86	21.19	20.64
32	**75.32**	**71.75**	**78.58**	**71.54**	**42.71**	**30.75**
64	73.71	71.25	73.21	64.51	37.08	28.73

2. Both the classification (3) and the contrastive loss (4) provide appreciable improvements to ZeroAE by helping to clear-cut the decision boundaries. They increase the averaged macro F_1-score by 8.68% and 6.31% respectively.
3. The disentanglement loss also helps to increase the averaged macro F_1-score by about 2%, since it separates the label-relevant features from the irrelevant ones and so the classifier can better distinguish between different labels. Indeed, as shown in Figure 2, after enforcing disentanglement, the label-relevant features become less correlated with the label-irrelevant features.

Impact of the backbone PLMs: Now let us check whether the proposed ZeroAE framework is agnostic to the backbone PLMs. Indeed, we replace BERT and GPT-2 with ERNIE and Chinese-GPT when tackling the customer complaints data in the previous subsection, and the results are still the best among the existing methods. Here we further replace the two BERTs and the GPT with the encoders and decoders in BART (Lewis et al., 2019), and the results are presented in Table 5. As expected, it can be observed that changing the backbone PLMs does not affect the superior performance of ZeroAE.

Impact of the number of fixed layers in the PLMs: We further investigate the influence of the number fixed layers in the PLMs on the performance of ZeroAE. The results are summarized in Table 6. We can find that setting the number of fixed layers to be 10, 0, and 6 respectively in the label-relevant encoder, the label-irrelevant encoder, and the decoder yields the highest averaged macro F_1-score. Therefore, we follow this setting in our experiments.

Impact of the IUAS threshold τ: We also provide experimental results for different choices of τ, namely, $\tau \in \{0.75, 0.8, 0.85, 0.9\}$. $\tau = 0.8$, which is the default value used in our paper, produces the highest avarged macro F_1-score.

Impact of the codebook size: Lastly, we conduct an empirical investigation to examine the impact of codebook size on the performance of our model. We perform several experiments with varying codebook sizes and present the results in Table 8. Based on the table, we observe that the parameter value of 32, as suggested in this paper, outperforms the other two sizes. This finding suggests that a codebook size that is too small may not provide adequate diversity to the decoder in ZeroAE, while a size that is too large may be superfluous for the given datasets.

4 Conclusion

In this paper, we propose a PLM-based autoencoder named ZeroAE for zero-shot text classification. The autoencoder framework enables the pretrained encoder and decoder to further complement and promote each other. Furthermore, the proposed IUAS training algorithm helps ZeroAE to deal with unlabeled data. Experiments on real-world datasets demonstrates that ZeroAE provides a much better solution to the domain adaptation (i.e., label-partially-unseen) and task adaptation (i.e., label-fully-unseen) problems in comparison with the SOTA methods.

5 Limitations

There are two limitations of our work: 1) As the overall loss function (12) comprises five components, we propose to directly add these components together. Although this simple summation already yields better results than the SOTA methods, we believe that it is better to tune the weights of these components based on expert knowledge, empirical experiments, or other machine-learning techniques. 2) In ZeroAE, we use three PLMs, including two BERTs and a GPT-2. Moreover, contrastive learning typically requires a relatively large batch size in order to collect a sufficient number of negative samples and achieve satisfying performance (Chen et al., 2020). The batch size in our experiments is typically 32. As a result, training ZeroAE incurs relatively large resource cost. In practice, we find that using four NVIDIA TESLA V100 GPUs with 32G memory works well, and further reducing the resources hurts the performance.

6 Ethical Considerations

We consider four datasets in our experiments, including Topic, Situation, Emotion, and Complaint. The first three are publicly accessible. The last one will be released upon publication. In particular for this dataset, 1) it does not contain any Personal Identifiable Information (PII); 2) This dataset is desensitized and encrypted; 3) Adequate data protection was carried out during the experiment to prevent the risk of data copy leakage, and the dataset was destroyed after the experiment; 4) This dataset is only used for academic research, and it does not represent any real business situation.

References

Eduardo Aguilar, Bhalaji Nagarajan, Rupali Khantun, Marc Bolaños, and Petia Radeva. 2021. Uncertainty-aware data augmentation for food recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4017–4024. IEEE.

Alexandre Alcoforado, Thomas Palmeira Ferraz, Rodrigo Gerber, Enzo Bustos, André Seidel Oliveira, Bruno Miguel Veloso, Fabio Levy Siqueira, and Anna Helena Reali Costa. 2022. Zeroberto: Leveraging zero-shot text classification by topic modeling. In *International Conference on Computational Processing of the Portuguese Language*.

Yashas Annadani and Soma Biswas. 2018. Preserving semantic relations for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7603–7612.

Iz Beltagy, Arman Cohan, Robert Logan IV, Sewon Min, and Sameer Singh. 2022. Zero-and few-shot nlp with pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 32–37.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Qi Chen, Wei Wang, Kaizhu Huang, and Frans Coenen. 2021. Zero-shot text classification via knowledge graph embedding for social media data. *IEEE Internet of Things Journal*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2022. Zerogen⊖+: Self-guided high-quality data generation in efficient zero-shot learning. *arXiv preprint arXiv:2205.12679*.

Rui Gao, Xingsong Hou, Jie Qin, Jiaxin Chen, Li Liu, Fan Zhu, Zhao Zhang, and Ling Shao. 2020. Zero-vae-gan: Generating unseen features for generalized and transductive zero-shot learning. *IEEE Transactions on Image Processing*, 29:3665–3680.

Ariel Gera, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein-Dor, and Noam Slonim. 2022. Zero-shot text classification with self-training. In *Conference on Empirical Methods in Natural Language Processing*.

He Huang, Changhu Wang, Philip S Yu, and Chang-Dong Wang. 2019. Generative dual adversarial network for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 801–810.

Haozhe Ji and Minlie Huang. 2021. Discodvt: Generating long text with discourse-aware discrete variational transformer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4224.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR.

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

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699.

Jin Li, Xuguang Lan, Yang Liu, Le Wang, and Nanning Zheng. 2019. Compressing unknown images with product quantizer for efficient zero-shot classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5472.

Hui Liu, Danqing Zhang, Bing Yin, and Xiaodan Zhu. 2021a. Improving pretrained models for zero-shot multi-label text classification through reinforced label hierarchy reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1051–1062.

Tengfei Liu, Yongli Hu, Junbin Gao, Yanfeng Sun, and Baocai Yin. 2021b. Zero-shot text classification with semantically extended graph convolutional network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8352–8359. IEEE.

Yang Liu, Quanxue Gao, Jin Li, Jungong Han, Ling Shao, et al. 2018. Zero shot learning via low-rank embedded semantic autoencoder. In *IJCAI*, pages 2490–2496.

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 preprint arXiv:1907.11692*.

Tingting Ma, Jin-Ge Yao, Chin-Yew Lin, and Tiejun Zhao. 2021. Issues with entailment-based zero-shot text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 786–796.

Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A Murthy. 2018. A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2188–2196.

Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, Xi-Zhao Wang, and QM Jonathan Wu. 2022. A review of generalized zero-shot learning methods. *IEEE transactions on pattern analysis and machine intelligence*.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*.

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

Shafin Rahman, Salman Khan, and Nick Barnes. 2019. Transductive learning for zero-shot object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6082–6091.

Oscar Sainz and German Rigau. 2021. Ask2transformers: Zero-shot domain labelling with pretrained language models. In *Proceedings of the 11th Global Wordnet Conference*, pages 44–52.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951.

Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. 2019. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8247–8255.

Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. 2021. Taxoclass: Hierarchical multi-label text classification using only class names. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4239–4249.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*, pages 1–40.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Congcong Wang, Paul Nulty, and David Lillis. 2022. Using pseudo-labelled data for zero-shot text classification. In *International Conference on Applications of Natural Language to Information Systems*, pages 35–46. Springer.

Qian Wang and Toby Breckon. 2020. Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6243–6250.

Xuesong Wang, Chen Chen, Yuhu Cheng, and Z Jane Wang. 2016. Zero-shot image classification based on deep feature extraction. *IEEE Transactions on Cognitive and Developmental Systems*, 10(2):432–444.

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021a. X-class: Text classification with extremely weak supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021b. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*.

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.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, pages 1112–1122. Association for Computational Linguistics (ACL).

Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. 2019. f-vaegan-d2: A feature generating framework for any-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10275–10284.

Peng Xu, Jackie Chi Kit Cheung, and Yanshuai Cao. 2020. On variational learning of controllable representations for text without supervision. In *International Conference on Machine Learning*, pages 10534–10543. PMLR.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *arXiv preprint arXiv:2202.07922*.

Zhiquan Ye, Yuxia Geng, Jiaoyan Chen, Jingmin Chen, Xiaoxiao Xu, Suhang Zheng, Feng Wang, Jun Zhang, and Huajun Chen. 2020. Zero-shot text classification via reinforced self-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3014–3024.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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 3914–3923.

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of NAACL-HLT*, pages 1031–1040.

Ling Zhang, Xiaosong Wang, Dong Yang, Thomas Sanford, Stephanie Harmon, Baris Turkbey, Bradford J Wood, Holger Roth, Andriy Myronenko, Daguang Xu, et al. 2020. Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. *IEEE transactions on medical imaging*, 39(7):2531–2540.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2856–2878.

Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. 2020. Don't even look once: Synthesizing features for zero-shot detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11693–11702.

A Related Works

In this section, we first provided a more detailed review of the literature on zero-shot text classification. Next, since autoencoders have already been used for zero-shot image classification, we further review autoencoder-based approaches in this field.

A.1 Zero-shot text classification

We hereby review the aforementioned two approaches for PLM-based zero-shot text classification: encoder-based and decoder-based models.

Encoder-based Methods (a.k.a discriminative or embedding-based methods) typically learn a projection to associate the texts and labels via BERT or RoBERTa (Liu et al., 2019). Some attempts have been made to formulate the text-label pair as a text entailment (TE) representation (Yin et al., 2019; Sainz and Rigau, 2021; Alcoforado et al., 2022), and the [CLS] token is then used to evaluate their similarity. Alternatively, the relation between the text-label pair can also be formulated as question-answering (QA) tasks (Puri and Catanzaro, 2019; Zhong et al., 2021). The corresponding classification results depend on the answer "yes/no" to the question of whether the text belongs to a certain category. These methods take advantage of the semantic correlation between texts and labels implied by BERT (Yin et al., 2019), but they may fail to adapt to different domains and tasks without labeling information (Ma et al., 2021). To alleviate this difficulty, one may introduce additional information such as knowledge graphs (Liu et al., 2021a,b) and label semantic information (Zhang et al., 2019).

Another problem with these methods is that they ignore the unlabeled data which may help to transfer knowledge from seen domains or targets to unseen ones. To take unlabeled data into account, self-training is typically employed (Ye et al., 2020; Chen et al., 2021; Gera et al., 2022). These methods iteratively use the BERT-based classifier to pseudo-label the unlabeled data and further use the pseudo-labels with high confidence to train the classifier. These methods can even be extended to tackle the label-fully-unseen scenario (Wang et al., 2021a; Shen et al., 2021; Wang et al., 2022), resulting in a general-purpose zero-shot learner for novel tasks. Unfortunately, the issue that impedes the use of self-training is error accumulation. The mistakenly pseudo-labeled data with high confidence could continuously bias the classifier and lead to inaccurate estimates. In this work, we instead propose an indirect uncertainty-aware sampling (IUAS) method to counteract this problem.

Decoder-based Methods (a.k.a generative-based methods) address the zero-shot text classification problem from another perspective. By utilizing the data synthesis power of GPT-2 (Radford et al., 2019), they either simulate texts for labels corresponding to unseen classes and tasks (Ye et al., 2022) or generate labels for unlabeled data (Schick and Schütze, 2021), and then train a classifier based on the generated data. However, the potentially low quality of the generated data may harm the classifier. This problem can be alleviated by using a larger PLM such as GPT-3 (Brown et al., 2020; Wang et al., 2021b) or conducting generated data selection with a noise-robust framework (Gao et al., 2022). Nonetheless, in all these approaches, the PLMs cannot be fine-tuned to be domain or task-specific. In this paper, we propose an autoencoder framework to complement encoder-based and decoder-based models, thus rendering BERT and GPT-2 to self-adapt to the domain or task at hand.

A.2 Autoencoder-based Zero-Shot Image Classification

Zero-shot learning based on autoencoder has seen success in the field of image classification, since it provides a framework to train the encoder and the decoder simultaneously for automatic domain adaption, while being able to tackle unlabeled data (Pourpanah et al., 2022). Thus, we provide a brief review here.

There are broadly three strategies for autoencoder-driven zero-shot image classification. The first one seeks to learn a better encoder for the sake of classification with the support of the decoder (Wang et al., 2016; Annadani and Biswas, 2018; Liu et al., 2018; Li et al., 2019). As pointed out in (Annadani and Biswas, 2018), the introduction of the decoder and the corresponding reconstruction loss improves the modeling capability of the encoder and the zero-shot recognition performance. On the other hand, the second strategy concentrates more on exploiting the decoder to generate samples for the unseen classes given the class attributes (Mishra et al., 2018; Xian et al., 2019; Huang et al., 2019; Gao et al., 2020; Zhu et al., 2020). Conditional variational autoencoders (CVAE) are typically used: the role of the encoder is to adapt the latent space in CVAE to the domain at hand, and therefore, facilitates the decoder to synthesize data for this domain. Different from the above two strategies, the third one (Schonfeld et al., 2019) first constructs two VAEs respectively for the image features and the class attributes and then aligns the two latent spaces via a cross-alignment loss. As such, the image and semantic features are projected into the same latent space that can be further utilized for classifying samples in the unseen classes.

Unfortunately, the key bottleneck with the above-mentioned methods is that the learned latent space, which is used for classification, is often corrupted by the label-irrelevant information that adversely affects the classification performance. One remedy to this problem is to separate the label-relevant from the label-irrelevant information by promoting disentanglement between them in the latent space (Schonfeld et al., 2019; Xian et al., 2019; Gao et al., 2020). In our work, we disentangle the two types of information via a discriminator in a similar fashion to (Schonfeld et al., 2019), due to its advantageous performance as shown in (Kim and Mnih, 2018).

B Datasets

We demonstrate the advantages of ZeroAE on four real-world datasets, including "Topic", "Situation", "Emotion" and "Complaint". The first three datasets are often used for benchmarking different zero-shot text classification approaches (Yin et al., 2019), and the last one aims to find the response team that can deal with the customer complaints reported to Ant Group based on the content of the complaints. Note that in the label-partially-unseen case, the two different versions of the first three datasets are provided in (Yin et al., 2019) with non-overlapping labels, in order to prevent the models from overfitting to some labels. The detailed statistics regarding how the datasets are split into training, validation, and testing sets are summarized in Table 1. More information on the four datasets is provided below:

1. **Topic Categorization**: The dataset contains Yahoo news articles with 10 topics, including "Society & Cultur","Health", "Computers & Internet", "Business & Finance", "Family & Relationships", "Science & Mathematics", "Education & Reference", "Sports", "Entertainment & Music", and "Politics & Government". The objective is to predict the topic given the news. The version v0 selects the first five classes as the seen classes, while the version v1 selects the other five classes.
2. **Situation Detection**: The dataset aims to find the type of an event, including the need situations (e.g., the need for water or medical aid) and the issue situations (e.g., crime violence), given the corresponding news. There are 12 classes in total, that is, "Regime change", "Crime violence", "Medical assistance", "Water supply", "Search/rescue", "Infrastructure", "Shelter", "Utilities, energy, or sanitation", "Evacuation", "Food supply", "Terrisms", and "None". The version v0 chooses the first six classes as the seen classes, while the version v1 chooses the other five classes excluding the class "None". We further follow the settings in (Ye et al., 2020) to remove the texts with multiple labels in our experiments.
3. **Emotion Detection**: The task here is to detect the emotion of the posters from the texts such as tweets, fairy tales, and emotional events. This dataset involves nine types of emotions, that is, "Sadness", "Anger", "Fear", "Shame", "Love", "Joy", "Disgust", "Surprise", and "Guilt". The versions v0 and v1 respectively treat the first five and the last four classes as the seen classes. Note that this task is more difficult than the above two tasks, since different emotions are often correlated with each other (Gera et al., 2022). For example, "Guilt" and "Shame" are synonyms, but represent two distinct classes here.
4. **Customer Complaint Triage**: The objective here is to find the response team in Ant Group that can deal with the customer complaints given the corresponding texts. There are 241 classes in total, but only 12 of them are seen classes in practice. The number of samples is also very small in this dataset, since the services provided by this company are relatively stable and there are few customer complaints. As a result, this task is very challenging, due to the data scarcity and the large number of unseen classes.

Table 9: Prompt for each datasets.

Dataset	Prompt
Topic	The news with _ topic is:
Situation	The news with _ situation is:
Emotion	The news with _ emotion is:
Complaint	The customer complaints about _ is:

C Ablation Study

In this section, we elaborate on the experiment settings in the first ablation study:

- $-$**IUAS**: In this experiment, the proposed IUAS approach is not used for training ZeroAE. In other words, the unlabeled data are only used in the text reconstruction flow, and no generated data are used to train the label-relevant encoder and the classifier.
- $-$**IUAS+Pseudo labeling**: After ablating IUAS, we instead employ the pseudo labeling approach to tackle the unlabeled data. Specifically, the pseudo labels are retained to train the classifier when the probability that the unlabled text belongs to a class is larger than 0.8.
- $-$**Classification Loss**: In this experiment, we remove the classification loss (3) when training ZeroAE, and treat the labeled data as unlabeled data. Note that this setting is called label-fully-unseen TZSL in this work.
- $-$**Contrastive Loss**: The contrastive loss (4) is removed when training ZeroAE.
- $-$**Disentanglement Loss**: Both the discriminator loss (7) and disentanglement loss (8) are removed during the training process of ZeroAE. As a result, the two latent spaces are not guaranteed to be disentangled.

Table 10: Summary of notations.

Symbol	Type (Size)	Meaning
C	Constant	The overall number of classes (both seen and unseen)
K	Constant	The number of codewords in the codebook
D	Constant	The dimension of the encoder output
W	Constant	The size of the vocabulary for the GPT-2
c	Constant	The index of the class with the highest probability score for one input text
τ	Constant	The threshold for data selection in IUAS
β	Constant	The weight of the second term in the VQ loss
$\mathbb{D}^S$	Dataset	Seen dataset
$\mathbb{D}^U$	Dataset	Unseen dataset
$\mathbb{D}^L$	Dataset	Labeled dataset
$\mathbb{D}^G$	Dataset	Generated dataset
Enc^R	Module	The label-relevant encoder (i.e., BERT with the [CLS] token as the output)
Enc^I	Module	The label-irrelevant encoder (i.e., BERT with the mean pooling of the last layer as the output)
Cls	Module	The linear classifier
Dec	Module	The decoder (i.e., prompt-based GTP-2)
Disc	Module	The linear discriminator
Θ	Parameter	The parameters of the discriminator
Ψ	Parameter	The parameters of all modules in ZeroAE except the discriminator
$\mathcal{H}$	Function	Cross entropy loss function
sg	Function	Stop gradient operation
f	Function	The function that converts a label to a one-hot vector
g	Function	The prompting function
q	Function	The density function of the latent variables
$\boldsymbol{x}_i$	Text	The i-th input text of the dataset
$\boldsymbol{x}'_i$	Text	The augmented text by applying EDA to $\boldsymbol{x}_i$
$\hat{\boldsymbol{x}}_i$	Text	The reconstructed or generated text
x_{ij}	Text	The j-th word in text $\boldsymbol{x}_i$
$\hat{x}_{ij}$	Tensor (W)	The vector of the probabilities that the word j in text $\hat{\boldsymbol{x}}_i$ takes the values in a predefined vocabulary
$\boldsymbol{y}_i$	Text	The true label of the i-th input text
$\hat{\boldsymbol{y}}_i$	Text	The predicted label for i-th input text
$\mathrm{y}_{1:C}$	Set	The label names for all C classes (both seen and unseen)
y_j	Text	The label name for the j-th class
$\boldsymbol{s}_i^R$	Tensor (C)	The classification probability vector for the input text $\boldsymbol{x}_i$
s_{ij}^R	Constant	The classification probability score of the label being y_j for input text $\boldsymbol{x}_i$
$\boldsymbol{e}$	Tensor $(K \times D)$	The VQ codebook
$\boldsymbol{e}_k$	Tensor (D)	The k-th codeword in the codebook
$\boldsymbol{z}_{ij}^R$	Tensor (D)	The label-relevant embedding after packing the input text $\boldsymbol{x}_i$ and the label name y_j under the TE framework
$\boldsymbol{z}_{ic}^{R\prime}$	Tensor (D)	The label-relevant embedding corresponding to the label y_c with the highest classification score
$\boldsymbol{z}_i^R$	Tensor $(C \times D)$	The label-relevant embedding for all C classes
$\boldsymbol{z}_i^I$	Tensor (D)	The label-irrelevant embedding for the input text $\boldsymbol{x}_i$
$\boldsymbol{z}^+$	Tensor $(C + 2D)$	The positive samples for the discriminator
$\boldsymbol{z}^-$	Tensor $(C + 2D)$	The negative samples for the discriminator

Algorithm 1 IUAS based Training Procedure for ZeroAE

```
Require: Texts and labels for the seen classes $\mathbb{D}^S = \{(\boldsymbol{x}_i^S, \boldsymbol{y}_i^S)\}$ if available, texts for the unseen classes $\mathbb{D}^U = \{(\boldsymbol{x}_i^U)\}$, label names for all $C$ classes $\mathrm{y}_{1:C} = \{\mathrm{y}_1, \cdots, \mathrm{y}_C\}$, and the IUAS threshold $\tau$;
Initialize the generated data as an empty set $\mathbb{D}^G = \{\}$;
repeat
    if $\mathbb{D}^G$ is not empty then
        Randomly pick $CK$ samples from $\mathbb{D}^G$ and denote the sample set as $\mathbb{D}^{G'}$;
    else
        $\mathbb{D}^{G'} = \{\}$;
    end if
    The labeled dataset can be computed as $\mathbb{D}^L = \mathbb{D}^S \cup \mathbb{D}^{G'}$;
    for $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ in $\mathbb{D}^L$ do        ▷ Text Reconstruction Flow for Labeled Data
        Augment $\boldsymbol{x}_i$ via EDA to obtain $\boldsymbol{x}_i'$
        for $\mathrm{y}_j$ in $\mathrm{y}_{1:C}$ do
            Pack $\boldsymbol{x}_i$ and $\mathrm{y}_j$ together following the TE framework "[CLS] $\boldsymbol{x}_i$ [SEP] hypothesis of $\mathrm{y}_j$ [SEP]";
            Encode the above packed text using the label-relevant encoder to obtain $z_{ij}^R$ following Eq. (1);
        end for
        Calculate the classification loss following Eq. (2)-(3);
        Pack $\boldsymbol{x}_i'$ and $\hat{\boldsymbol{y}}_i$ following the TE framework as the positive sample to obtain $z_{ic}^{R'}$;
        Calculate the contrastive loss following Eq. (4);
        Obtain the label-irrelevant latent variable $\boldsymbol{z}_i^I$ following Eq. (5) and calculate the VQ loss following Eq. (6);
        Calculate the disentanglement loss following Eq. (8);
        Reconstruct the text $\hat{\boldsymbol{x}}_i$ following Eq. (9) and calculate the reconstruction loss following Eq. (10);
    end for
    for $(\boldsymbol{x}_i)$ in $\mathbb{D}^U$ do        ▷ Text Reconstruction Flow for Unlabeled Data
        Augment $\boldsymbol{x}_i$ via EDA to obtain $\boldsymbol{x}_i'$;
        for $\mathrm{y}_j$ in $\mathrm{y}_{1:C}$ do
            Pack $\boldsymbol{x}_i$ and $\mathrm{y}_j$ together following the TE framework and obtain $z_{ij}^R$ following Eq. (1);
        end for
        Calculate the classification score following Eq. (2);
        Pack $\boldsymbol{x}_i'$ and $\hat{\boldsymbol{y}}_i$ following the TE framework as the positive sample to obtain $z_{ic}^{R'}$;
        Calculate the contrastive loss following Eq. (4);
        Obtain the label-irrelevant latent variable $\boldsymbol{z}_i^I$ following Eq. (5) and calculate the VQ loss following Eq. (6);
        Calculate discriminator loss following Eq. (8);
        Reconstruct the text $\hat{\boldsymbol{x}}_i$ following Eq. (9) and calculate the reconstruction loss following Eq. (10);
    end for
    Fix the discriminator and update the remaining parts of ZeroAE by minimizing Eq. (12) using gradient descent;
    for $\mathrm{y}_j$ in $\mathrm{y}_{1:C}$ do        ▷ Train Discriminator
        Randomly pick positive sample and negative sample as in §2.3, and calculate the discriminator loss following Eq. (7);
    end for
    Update the discriminator only by minimizing Eq. (7) using gradient descent;
    for $\mathrm{y}_j$ in $\mathrm{y}_{1:C}$ do        ▷ Text Generation in Label Reconstruction Flow
        for $\boldsymbol{e}_k$ in $\boldsymbol{e}$ do
            Generate data using the decoder $\hat{\boldsymbol{x}} = \mathrm{Dec}(h(\mathrm{y}_j), \boldsymbol{e}_k)$;
            Put $(\hat{\boldsymbol{x}}, \mathrm{y}_j)$ in the generated dataset $\mathbb{D}^G$;
        end for
    end for
    for $(\boldsymbol{x}_i)$ in $\mathbb{D}^U$ do        ▷ Data Selection in Indirect Uncertainty-Aware Sampling (IUAS)
        for $\mathrm{y}_j$ in $\mathrm{y}_{1:C}$ do
            Pack $\boldsymbol{x}_i$ and $\mathrm{y}_j$ together following the TE framework and obtain $\boldsymbol{z}_{ij}^R$ following Eq. (1);
        end for
        Calculate the classification score following Eq. (2);
        if $\max \boldsymbol{s}_i^R > \tau$ then
            Remove $\boldsymbol{x}_i$ from $\mathbb{D}^U$;
        end if
    end for
until the maximum number of epochs is reached or early stop criteria is met.
```

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 5.

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 2, 3.

☑ B1. Did you cite the creators of artifacts you used?
Section 1, 2, 3.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 7.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
In table 1.

C ☑ Did you run computational experiments?

Section 4.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 2 and Section 5.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4, Appendix C and Table 6

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.