REGEN: Zero-Shot Text Classification via Training Data Generation with Progressive Dense Retrieval

Yue Yu¹, Yuchen Zhuang¹, Rongzhi Zhang¹, Yu Meng², Jiaming Shen³, Chao Zhang¹
¹ Georgia Institute of Technology, GA, USA
² University of Illinois at Urbana-Champaign, IL, USA
³ Google Research, NY, USA
{yueyu, yczhuang, rongzhi.zhang, chaozhang}@gatech.edu
yumeng5@illinois.edu, jmshen@google.com

Abstract

With the development of large language models (LLMs), zero-shot learning has attracted much attention for various NLP tasks. Different from prior works that generate training data with billion-scale natural language generation (NLG) models, we propose a retrieval-enhanced framework to create training data from a general-domain unlabeled corpus. To realize this, we first conduct contrastive pretraining to learn an unsupervised dense retriever for extracting most relevant documents using class-descriptive verbalizers. We then further propose two simple strategies, namely Verbalizer Augmentation with Demonstrations and Self-consistency Guided Filtering to improve the topic coverage of the dataset while removing noisy examples. Experiments on nine datasets demonstrate that REGEN achieves 4.3% gain over strongest baselines and saves around 70% of the time when compared with baselines using large NLG models. Besides, REGEN can be naturally integrated with recently proposed large language models to boost performance¹.

1 Introduction

Text classification serves a fundamental task in Natural Language Processing (NLP) with a broad spectrum of applications. Recently, large pretrained language models (PLMs) (Devlin et al., 2019) have achieved strong performance on text classification with a large amount of task-specific training data. However, in real world scenarios, collecting labeled data can be challenging due to the cost of time, money, and domain expertise.

To reduce the burden of human annotation, we study automatic dataset generation for text classification under the zero-shot setting, where no task-specific or cross-task data is available. Such a setting is different from previous works that use a large collection of labels from auxiliary tasks for zero-shot text classification (Yin et al., 2019; Gera et al., 2022; Wei et al., 2022; Sanh et al., 2022), and is particularly challenging since we need to adapt the language understanding abilities of PLMs to target classification tasks with minimal supervision.

Prior works on zero-shot synthetic dataset generation mainly fall into two categories: (1) Generative methods leverage a billion-scale NLG model to generate class-conditioned texts for PLM fine-tuning (Meng et al., 2022; Ye et al., 2022a,b). While these methods work well on easy tasks (e.g. binary classification), they can be fragile on challenging tasks with more classes, as the generated text can be less discriminative. Besides, the gigantic size of the NLG model will also cause the inefficiency issue. (2) Mining-based methods design rule-based regular expressions to extract text from the background corpus as synthesized training data (van de Kar et al., 2022), but these rules are often too simple to capture the complex semantics of text. As a result, the mined dataset contains many incorrectly-labeled data, and the fine-tuned PLM can easily overfit noisy labels.

We design a new framework REGEN² to solve zero-shot text classification. The setting of REGEN is close to the mining-based technique (van de Kar et al., 2022), where a set of class-specific verbalizers and a collection of general-domain unlabeled corpus are available. Motivated by the limitation of hard matching with regular expressions which hardly preserves the meaning of verbalizers, we propose to leverage dense retrieval (DR) (Lee et al., 2019; Karpukhin et al., 2020; Xiong et al., 2021; Sun et al., 2022a; Cui et al., 2022), which calculates semantic relevance in a continuous representation space, for dataset curation. With such a soft matching mechanism, DR is able to better encode the category-specific semantics and thus fetch the relevant documents from the corpus. To integrate

¹The code and unlabeled corpus will be released in https://github.com/yueyu1030/ReGen.
²Retrieval-Enhanced Zero-shot Data Generation.
DR with the target classification task, we employ two PLMs: one retrieval model \( R_\theta \) to extract the most relevant documents from the unlabeled corpus for synthetic dataset curation, and one classification model \( C_\phi \) to be fine-tuned on the generated synthetic dataset to perform the downstream task. Before performing text retrieval, we first conduct contrastive learning on the unlabeled corpus to further pretrain the retrieval model \( R_\theta \) for producing better sequence embeddings. Then, with the retrieval model, we use the verbalizers from each class as queries to retrieve relevant documents from the unlabeled corpus, which will be used as the training data for target tasks.

Simply fine-tuning the classifier on the above training data may yield limited performance, as the verbalizers are often too generic to cover all the category-related topics (e.g., the word ‘sports’ alone does not cover concrete types of sports). Thus, the retrieved data may contain noisy and irrelevant documents. To enhance the quality of the synthetic dataset, we conduct multi-step retrieval with two additional strategies to strengthen our framework: (1) we augment the verbalizer with the retrieved documents from the previous round as additional information (Xu and Croft, 2017) to enrich its representation, which allows for extracting more relevant documents for the downstream task. (2) we exploit self-consistency to filter the potentially incorrect examples when the pseudo labels produced by the retrieval model \( R_\theta \) and the classifier \( C_\phi \) disagree with each other. We note that \textsc{ReGen does not} use annotated labels from any other tasks, making it applicable to the true zero-shot learning. Besides, \textsc{ReGen} only requires two BERT\textsubscript{base} scale PLMs, which is efficient compared with methods using large NLG models.

Our contribution can be summarized as follows: (1) We propose \textsc{ReGen}, a framework for zero-shot dataset generation with a general-domain corpus and retrieval-enhanced language models. (2) We develop two additional techniques, namely verbalizer augmentation with demonstration and self-consistency guided filtering to improve the quality of the synthetic dataset. (3) We evaluate \textsc{ReGen} on nine NLP classification tasks to verify its efficacy. We also conduct detailed analysis to justify the role of different components as well as the robustness of \textsc{ReGen} over different verbalizers.

2 Related Work

Zero-shot Text Classification (ZSTC) aims to categorize the text document without using task-specific labeled data. With pretrained language models, a plenty of works attempted to convert the classification task into other formats such as masked language modeling (Hu et al., 2022; Gao et al., 2021), question answering (Zhong et al., 2021; Wei et al., 2022; Sanh et al., 2022) or entailment (Yin et al., 2019; Gera et al., 2022) for zero-shot learning. These works are orthogonal to \textsc{ReGen} as we do not directly perform inference and do not leverage human annotations from additional tasks.

More relevant to us, there are some recent studies that perform ZSTC via generating a task-specific dataset using NLG models, which is then used to fine-tune a classifier for the target task such as text classification (Ye et al., 2022a,b; Meng et al., 2022), sentence similarity calculation (Schick and Schütze, 2021b), commonsense reasoning (Yang et al., 2020; Kan et al., 2021), and instruction-based tuning (Wang et al., 2022). Unfortunately, the generation step is time-consuming and the quality of the generated text can be less satisfactory in capturing fine-grained semantics. The most relevant work to us is (van de Kar et al., 2022), which also extracts documents from the unlabeled corpus to form the training set. But it simply uses regular expressions to mine documents and cannot fully capture the contextual information of verbalizers. Instead, we leverage dense retrieval for concept understanding and obtain the most relevant documents, which is combined with verbalizer augmentation to improve retrieval quality.

On the other hand, retrieval-augmented language models have been used in language modeling (Khandelwal et al., 2020; Borgeaud et al., 2022), OpenQA (Jiang et al., 2022; Sachan et al., 2021), information extraction (Zhuang et al., 2022) and knowledge-intensive tasks (Lewis et al., 2020; Izacard et al., 2022b), where tokens or documents are retrieved based on contextual representations and are used as additional inputs to support target tasks. While such a paradigm has also been explored for zero-shot learning, it is mainly used for zero-shot prompt-based inference (Shi et al., 2022; Chen et al., 2022). Instead, we empirically demonstrate the efficacy of retrieval-enhanced learning for zero-shot dataset curation with an unsupervised dense retrieval model.
3 Preliminaries

◊ Setup. We focus on synthesizing a task-specific dataset for text classification (Meng et al., 2022; van de Kar et al., 2022). Besides, we stick to the strict zero-shot setup (Perez et al., 2021), where no labeled examples from either target tasks or other tasks are available.

◊ Available Resources. Besides annotated labels, the availability of massive task-specific unlabeled data is also a rarity — in prior works, such unlabeled data is obtained via removing the ground-truth label from the original dataset (Meng et al., 2020b), and can be scarce in real zero-shot settings (Tam et al., 2021). The most accessible information is a collection of general-domain unlabeled corpus $D$ (e.g., Wiki), which is freely available online and has been used for pretraining (Devlin et al., 2019; Gururangan et al., 2020). Recent works also use such an external corpus for zero-shot learning (Shi et al., 2022; van de Kar et al., 2022).

◊ Task Formulation. With the above discussion, we consider the classification task where we are given the label set $Y = \{1, 2, \ldots, c\}$ ($c$ is the number of classes), and a mapping $M: Y \rightarrow Y$ that converts each label $y \in Y$ into a class-descriptive verbalizer $w_y \in W$. We also assume a general-domain unlabeled corpus $D$ is available. We seek to curate training data $T$ from $D$ and learn a PLM $G$ which will be fine-tuned as the classifier.

◊ Backgrounds for Dense Retrieval (DR). In dense retrieval (Lee et al., 2019), the PLM is used to represent queries and documents in dense vectors. The relevance score $f(q, d)$ is calculated with a scoring function (e.g., dot product) between query and document vectors

$$f(q, d) = \text{sim}(R_\theta(q), R_\theta(d)), \tag{1}$$

where the embedding of the [CLS] token from the final layer of $R_\theta$ is used as the representation for both queries and documents. In practice, the documents are encoded offline, and can be efficiently retrieved using approximate nearest neighbor search (ANN) with the queries (Johnson et al., 2021).

4 Method

In this section, we present ReGEN (our framework) and introduce the major components.

4.1 Contrastive Pretraining for Retriever $R_\theta$

Directly using BERT for retrieval can lead to unsatisfactory results since BERT embeddings are not tailored for retrieval application (Gao et al., 2021b). To effectively train a dense retrieval model without relevance supervision, we hypothesize that two sentences from the same document share similar semantics as they may describe the same topic. Then, we continuously pretrain the PLM on the corpus $D$ with contrastive learning (Gao and Callan, 2022; Izacard et al., 2022a; Yu et al., 2022b): Given a document $d_i \in D$, the positive pair $(x_i, x_i^+)$ is constructed by randomly sampling two disjoint sentences from $d_i$. Let $h_i = R_\theta(x_i), h_i^+ = R_\theta(x_i^+)$ denote the representation of $x_i$ and $x_i^+$ encoded by the retriever $R_\theta$, the training objective of contrastive learning for pair $(x_i, x_i^+)$ with a mini-batch of $N$ pairs is:

$$\ell_{ct} = -\log \frac{e^{\text{sim}(h_i, h_i^+)/\tau}}{\sum_{j=1}^{N} e^{\text{sim}(h_i, h_j^+)/\tau}}, \tag{2}$$

where we use in-batch instances as negative samples (Gillick et al., 2019), $\text{sim}(h_i, h_j^+) = h_i^\top h_j^+$ is the dot product, and $\tau = 1$ is the parameter for temperature. Contrastive learning improves the representations by promoting the alignment of similar text sequences and the uniformity of unrelated text sequences, thus enhancing the embedding quality for documents in $D$.

4.2 Overall Pipeline

With a pretrained retrieval model $R_\theta$, ReGEN follows a retrieve-then-finetune pipeline to curate the training data from the corpus $D$ which will be used to finetune the PLM classifier $G$. The details of our framework are described as follows.

Document Retrieval with Verbalizers. With the class-specific verbalizers, we construct the input queries for each class to retrieve the relevant documents from $D$. Formally, the query for the $i$-th class ($1 \leq i \leq c$) can be expressed as

$$q_i = [\text{CLS}] \circ \mathcal{P}(w_i) \circ [\text{SEP}],$$

where $\mathcal{P}(w_i)$ is the template for the corresponding class with the verbalizer $w_i$ and $\circ$ stands for the concatenation operation. For instance, a query for the binary sentiment classification can be formulated as $q_i = [\text{CLS}]$. It was $w_i = [\text{SEP}]$, where $w_1$ and $w_2$ ($c = 2$ in this case) stand for the verbalizers, namely “bad” (negative) and “great” (positive), respectively. By feeding the class-dependent query into the retriever $R_\theta$, we expect the retriever to understand its contextualized semantics (Rubin et al., 2022), and extract the relevant documents from the corpus which serve as training examples for the cor-
Algorithm 1: Process of ReGEN.

\textbf{Input:} $D$: Unlabeled Corpus; $Y$: Label space; $P$: Verbalizers; $R_\theta$: Retrieval Model; $C_\phi$: Classification Model; $T$: Rounds of Retrieval.

\textbf{Step 0:} Contrastive Learning.

Pretrain $R_\theta$ with Contrastive Learning via Eq. 2.

\textbf{for} $t = 1, 2, \ldots, T$ \textbf{do}

\hspace{1cm} // Step 1: (Multi-step) Document Retrieval.

\hspace{2cm} if $t = 1$ then

\hspace{3cm} Retrieve Documents $T^1$ with $P$ via Eq. 3.

\hspace{2cm} else

\hspace{3cm} Retrieve Documents $T^t$ with $P$ and $T^{t-1}$ via Eq. 6. // Verbalizer Augmentation.

\hspace{2cm} \textbf{// Step 2: Document Filtering.}

\hspace{2cm} Obtain Filtered Dataset $\tilde{T}^t$ via Eq. 7.

\hspace{2cm} \textbf{// Step 3: Language Model Fine-tuning.}

\hspace{3cm} Fine-tune PLM $C_\phi^t$ with $\tilde{T}^t$ via Eq. 4.

\textbf{Output:} The dataset $\tilde{T}^t$ and the PLM classifier $C_\phi^t$.

responding category. For the $i$-th class, the initial retrieved dataset $T_i^1 \subset D$ can be written as

\[ T_i^1 = \text{Top-k } f(q_i, d), \tag{3} \]

where $f(q, d)$ is defined in Eq. 1. The full retrieved dataset can be expressed as $T_i^t = \bigcup_{1 \leq c \leq T} T_i^1$.

**Fine-Tuning PLM with Curated Data.** After obtaining the training data $T$ from the corpus\(^3\), one can fine-tune a PLM classifier $C_\phi$ for the downstream task. To achieve better fine-tuning stability and generalization, we adopt the simple label smoothing (LS) technique (Müller et al., 2019), which mixes the one-hot labels with uniform vectors. For a training example $(x, y) \in T$, $C_\phi$ is trained to minimize the divergence between the label and the classifier’s prediction $p_\phi(x)$ as

\[ \min_\phi \ell_{\text{th}} = -\frac{1}{c} \sum_{j=1}^{c} q_j \log(p_\phi(x)_j), \tag{4} \]

where $q_j = 1(y = j)(1-\alpha) + \alpha/c$ is the smoothed label and $\alpha = 0.1$ is the smoothing term. LS prevents $C_\phi$ from overfitting to training data by forcing it to produce less confident predictions.

**4.3 Progressive Training Data Curation via Multi-step Dense Retrieval**

Although the aforementioned pipeline can retrieve a set of documents used for training ($T^1$), the performance can still be suboptimal because (1) the training set only have limited coverage as the verbalizers only contains few key words which is too specific to fully represent the categorical information; (2) the training set still contain noisy or task-irrelevant documents as the $R_\theta$ may not always retrieve texts pertaining to the desired class. To overcome these drawbacks, we perform document retrieval for multiple rounds, employing two additional strategies as described below.

**Verbalizer Augmentation with Demonstrations.** The verbalizers often contain only a few words and are insufficient to perfectly reflect the underlying information. Motivated by the recently proposed demonstration-based learning (Brown et al., 2020; Min et al., 2022) which augments the input with labeled examples to support in-context learning, we aim to enrich verbalizers with top retrieved documents for improving their representations (Yu et al., 2021), and thus enhancing the quality of the retrieved data. Specifically, in the $t$-th ($t > 1$) round, we use the retrieved documents from the $t$-1 round as demonstrations to augment the verbalizer for the $i$-th class as\(^4\)

\[ q_{i,j}^t = [\text{CLS}] \circ \mathcal{P}(w_i) \circ [\text{SEP}] \circ d_{i,j}^{t-1} \circ [\text{SEP}], \tag{5} \]

where $d_{i,j}^{t-1}$ is the $j$-th documents for the $i$-th class in the previous dataset $\tilde{T}^{t-1}$. With the augmented queries, $T_i^t$ and $T^t$ are obtained via combining the retrieved documents as

\[ T_i^t = \bigcup_{i} \bigcup_{d \in D} (\text{Top-k } f(q_{i,j}^t, d))^t, \quad T^t = \bigcup_{1 \leq i \leq c} T_i^t. \tag{6} \]

**Filtering Noisy Data guided via Self-consistency.**

The above retrieval process may also introduce noisy examples due to the limited capability of the retrieval model. While the label smoothing in Eq. 4 can mitigate this issue during fine-tuning, it is a generic technique without considering task-specific knowledge. To further fulfill the denoising purpose, we simply leverage the classifier from the previous round and exploit the consistency between the retriever and classifier to identify potential incorrect examples. For the example from the $t$-th round ($t > 1$) denoted as $(x', y') \in T^t$ where $y'$ is the label for the augmented verbalizer, we generate the predicted label using the classifier $C_{\phi}^{t-1}$ from the previous round\(^5\) as $\hat{y}' = \argmax_{y'} p_{\phi}^{t-1}(x')$.

Then, the filtered dataset $\tilde{T}^t$ is expressed as

\[ \tilde{T}^t = \{ (x', y') \in T^t \mid \argmax_{y'} p_{\phi}^{t-1}(x') = y' \}. \tag{7} \]

To interpret Eq. 7, we only preserve examples

\[^4\]\(^5\)We omit multiple queries for each class after this step.

\[^5\]When $t = 1$, we use the zero-shot prompting model as the classifier due to the absence of the ‘previous model’.
where the prediction from the previous classifier \( \hat{y}^{t-1} \) and the retrieved label \( y^t \) are consistent to fine-tune the classifier \( C_{\phi} \), thus serving as an additional protection for \( C_{\phi} \) against overfitting to label noises.

### 4.4 Overall Algorithm

The procedure of REGEN is summarized in Algorithm 1. Note that the retrieval model pretraining and corpus indexing only need to be done once before applying to all datasets. In each round of retrieval, it only needs one extra ANN retrieval operation per query, which is efficiently supported by FAISS (Johnson et al., 2021). We conduct the efficiency study in the Section 5.9.

### 5 Experiments

#### 5.1 Experimental Setups

- **Datasets.** In this work, we select AG News (Zhang et al., 2015), DBPedia (Lehmann et al., 2015), Yahoo (Zhang et al., 2015) and NYT (Meng et al., 2020a) for topic classification, and IMDB (Maas et al., 2011), SST-2 (Socher et al., 2013), Amazon (McAuley and Leskovec, 2013), MR (Pang and Lee, 2005), Yelp (Zhang et al., 2015) for sentiment analysis. All the datasets are in English. We report performance on the test set when available, falling back to the validation set for SST-2. The details for these datasets can be found in table 1.

- **Corpus.** We follow (Shi et al., 2022; van de Kar et al., 2022) to obtain a heterogeneous collection of text that are broadly relevant to tasks in our experiments as the general-domain unlabeled corpus \( \mathcal{D} \). Specifically, we select Wiki (Petroni et al., 2021), subsets of Reviews (He and McAuley, 2016) and Real News (Zellers et al., 2019) to form the corpus. The detailed information and preprocessing steps for these corpora are shown in Appendix B.

#### Metrics.** We use F1 score as the metric for NYT as the label distribution is imbalanced. Accuracy is used for the remaining tasks.

#### Baselines.** We consider various baselines, including both zero-shot inference and dataset generation methods. Details of the baselines are in Appendix C. We also list the results with extra resources (e.g. large PLMs, task-specific samples, or knowledge bases), but only for reference purposes, since we do not claim REGEN achieves state-of-the-art performance on zero-shot text classification. Rather, we consider REGEN as a better approach to synthesizing datasets in a zero-shot manner for text classification tasks.

#### Implementation Details.** For implementation, we use PyTorch (Paszke et al., 2019) and HuggingFace (Wolf et al., 2019). We set the retrieval rounds \( T = 3 \), the \( k \) used in ANN in Eq. 3 to 100 for the 1st round and 20 for later rounds in Eq. 6. The number of the training data per class is set to no more than 3000 (Meng et al., 2022). Under the zero-shot learning setting, we keep all hyperparameters the same across all tasks due to the lack of validation sets. In principle, REGEN is compatible with any dense retriever \( R_{\theta} \) and classifier \( C_{\phi} \). In this work, we initialize \( R_{\theta} \) from Condenser (Gao and Callan, 2021) and fine-tune RoBERTa-base (Liu et al., 2019) as \( C_{\phi} \). See App. D for details.

### 5.2 Main Experiment Results

The results of REGEN and compared baselines on nine tasks are in Table 2. From these results, we have the following observations:

1. REGEN significantly surpasses fair baselines on average of nine datasets, and often achieves comparable or even better results against methods using extra task-specific information. Compared with our direct baseline (van de Kar et al., 2022) using regular expressions to mine training data, REGEN achieves 4.3% gain on average. The gain is more notable (6.8%) for topic classification with more classes. These results justify that dense retrieval serves as a more flexible way to understand the category and can extract training data being semantically closer to the target topics.

2. SuperGen (Meng et al., 2022) achieves strong results on sentiment tasks. However, its performance diminishes for multi-class topic classification, suggesting that NLG-based dataset generation methods may struggle to produce sufficiently accurate and distinct texts for fine-grained classification.

---

**Table 1: Dataset statistics.**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Task</th>
<th>Class</th>
<th># Test</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGNews</td>
<td>News Topic</td>
<td>4</td>
<td>7.6k</td>
<td>Accuracy</td>
</tr>
<tr>
<td>DBPedia</td>
<td>Wikipedia Topic</td>
<td>14</td>
<td>70k</td>
<td>Accuracy</td>
</tr>
<tr>
<td>Yahoo Topics</td>
<td>Web QA Topic</td>
<td>10</td>
<td>60k</td>
<td>Accuracy</td>
</tr>
<tr>
<td>NYT</td>
<td>News Topic</td>
<td>9</td>
<td>30k</td>
<td>F1</td>
</tr>
<tr>
<td>IMDB</td>
<td>Movie Review Sentiment</td>
<td>2</td>
<td>25k</td>
<td>Accuracy</td>
</tr>
<tr>
<td>MR</td>
<td>Movie Review Sentiment</td>
<td>2</td>
<td>2k</td>
<td>Accuracy</td>
</tr>
<tr>
<td>SST-2</td>
<td>Movie Review Sentiment</td>
<td>2</td>
<td>0.8k</td>
<td>Accuracy</td>
</tr>
<tr>
<td>Amazon</td>
<td>Product Review Sentiment</td>
<td>2</td>
<td>40k</td>
<td>Accuracy</td>
</tr>
<tr>
<td>Yelp</td>
<td>Restaurant Review Sentiment</td>
<td>2</td>
<td>38k</td>
<td>Accuracy</td>
</tr>
</tbody>
</table>

---

6We follow (Hu et al., 2022) to subsample a 40K subset from the original 400K test data for faster evaluations, which has little effect on the average performance in our pilot studies.
Table 2: Main results. We report average performance and standard deviation across 5 runs if fine-tuning is applied. ♯: concurrent work, ∗: use the same corpus and template as REGEN for fair comparisons, ‡: use labeled data from auxiliary tasks, †: use task-specific corpus, ±: use billion-scale PLMs, §: use additional knowledge base.

Table 3: Ablation Study. For w/o DC, we use $R_\theta$ to calculate similarity between samples and labels for zero-shot learning. For w/o MSR, we only retrieve the same size of data as REGEN for one round with verbalizers. For w/o LS, we use one-hot labels for fine-tuning.

(3) REGEN also delivers competitive performance against zero-shot learning and weakly-supervised text classification baselines without requiring additional resources, such as larger language models or task-specific unlabeled data. This suggests that dataset generation serves as an alternative approach for zero-shot text classification.

5.3 Ablation Studies

Effect of Different Components. Table 3 shows the result of ablation studies on four datasets, which demonstrates the superiority of retrieving texts from the corpus for training data creation as well as conducting multi-step retrieval. Besides, label smoothing also results in performance gain as it mitigates the effect of noisy labels for fine-tuning. Besides, we plot the result over different rounds of retrieval in Fig. 1. It is clear that both multi-step retrieval and filtering progressively enhance the performance of target tasks, justifying their necessity for improving the quality of training data. We have also attempted to conduct more retrieval rounds, but do not observe significant performance gains.

Study of Dense Retrievers. We compare the retrieval model $R_\theta$ with other off-the-shelf unsupervised retrieval models. Here we choose one sparse model BM25 (Robertson et al., 2004) and three DR models: Condenser (Gao and Callan, 2021), SimCSE (Gao et al., 2021b), and Contreiver (Izacard et al., 2022a). From Figure 2, we observe that the performance of BM25 is not satisfactory, since simply using lexical similarity is insufficient to retrieve a set of diverse documents for fine-tuning. Besides, our retrieval model outperforms other unsupervised DR models for two reasons: (1) Condenser and SimCSE are pretrained over short sentences, and the learning objective is suboptimal for long documents; (2) these models are not pretrained on the corpus used in our study and suffer from the distribution shifts (Yu et al., 2022b). Instead, our strategy can better adapt the PLM for the retrieval task.
In the following sections, we mainly compare ReGen with Mining (van de Kar et al., 2022) and SuperGen (Meng et al., 2022) as they are closest baselines to us.

5.4 Effect of the Amount of Generated Data
Figure 3 shows the results of using different amount of training data (after filtering). Overall, we find that the performance first improves significantly when the number of training data is small (e.g., 100), then becomes stable with more retrieved data. This is because with too many generated data, it may also introduce more label noise and reduce the quality of training data. Nevertheless, ReGen outperforms baselines under different volumes of training samples, justifying its advantage.

5.5 Fusing ReGen with Large Language Models (LLMs)
In this section, we give a simple demonstration of how to leverage recently-proposed large language models (e.g. GPT-4 (OpenAI, 2023)) to further boost the performance. As LLMs have demonstrated strong ability for text generation, we use them to augment the verbalizer before retrieving documents from the general-domain corpus. The details are in Appendix E.3.

From Table 4, we observe that expanded verbalizers lead to consistent performance gains on two datasets. Although the scale of the improvement is not that significant, it shows some effectiveness with such cheap plug-in techniques of using LLMs for boosting ReGen.

5.6 Using ReGen in Few-Shot Settings
ReGen can also be combined with a few labeled examples to improve the performance. We follow (Meng et al., 2022) to fine-tune $C_\phi$ with few-shot examples and the synthetic dataset (Details in Appendix E.1) using IMDB and AG News as examples. From Fig. 4, we observe that ReGen improves over the vanilla few-shot fine-tuning under all studied regions (32 to 512 labels per class), while baselines cannot further promote the performance with more training samples. Quantitatively, the performance of ReGen is equivalent to that of fine-tuning with 128-256 labeled documents per class. With 32 labels per class, ReGen achieves comparable performance of vanilla fine-tuning with 4x-8x labeled examples. These results verify that ReGen promotes label efficiency of PLMs.

5.7 Robustness over Different Verbalizers
As ReGen and zero-shot dataset generation methods always rely on a class-dependent verbalizer to steer the whole process, we study the impact of different verbalizers on the final performance. We use IMDB and AG News as two datasets, and create three different groups of verbalizers other than the default ones for comparison (Details in Appendix E.2). From Table 5, we observe that ReGen generally outperforms baselines on 7 out of 8 cases. ReGen also has lower performance variance across four groups of verbalizers. These results reveal that ReGen does not rely on specific designs of verbalizers, and are more robust over
different verbalizers.

5.8 The Effect on General-domain Corpus $D$

We study the effect of corpus $D$ by conducting retrieval on different subsets from $D$. As shown in Figure 5(a), we observe better performance when the corpus aligns with the target task well (e.g., News for AG News). This is expected as the model suffers less from the distribution shift issue. Besides, REGEN outperforms the mining method under all settings, justifying its superior ability to retrieve relevant text even if there is a domain mismatch between the task and corpus.

Fig. 5(b) exhibits the relation on the lexical similarity (measured by weighted Jaccard score), and the performance gap between REGEN and fully-supervised BERT (details in Appendix G). Overall, there is a negative correlation among performance gaps and the distribution similarities, as REGEN performs closer to fully-supervised models on tasks where task-specific documents share more similar lexical patterns with the general-domain corpus.

5.9 Efficiency Studies

Table 6 measures the efficiency of REGEN and baselines. While the pretraining and indexing corpus for REGEN can be time-consuming, it only needs to be done once, thus the overall running time of REGEN is significant lower than the baseline using large NLG models (Meng et al., 2022).

<table>
<thead>
<tr>
<th>Operation</th>
<th>Mining</th>
<th>SuperGen</th>
<th>REGEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretraining</td>
<td>—</td>
<td>—</td>
<td>23h</td>
</tr>
<tr>
<td>Indexing of Corpus/Per doc</td>
<td>—</td>
<td>—</td>
<td>6h/4ms</td>
</tr>
<tr>
<td>Curating Dataset Per Task</td>
<td>1.4h</td>
<td>20.4h</td>
<td>0.6h</td>
</tr>
<tr>
<td>Filtering Per Task</td>
<td>0.2h</td>
<td>0.1h</td>
<td>0.5h</td>
</tr>
<tr>
<td>Model Fine-tuning Per Task</td>
<td>0.4h</td>
<td>0.3h</td>
<td>0.7h</td>
</tr>
<tr>
<td>Total Time (for all Tasks)</td>
<td>10h</td>
<td>104h</td>
<td>38h</td>
</tr>
</tbody>
</table>

Table 7: Automatic evaluation results on three metrics.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metrics</th>
<th>Mining</th>
<th>SuperGen</th>
<th>REGEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>AG News</td>
<td>Correctness ($\uparrow$)</td>
<td>0.815</td>
<td>0.971</td>
<td>0.986</td>
</tr>
<tr>
<td></td>
<td>Diversity ($\downarrow$)</td>
<td>0.144</td>
<td>0.915</td>
<td>0.361</td>
</tr>
<tr>
<td></td>
<td>Distribution Sim. ($\downarrow$)</td>
<td>0.856</td>
<td>0.803</td>
<td>0.865</td>
</tr>
<tr>
<td>Sentiment</td>
<td>Correctness ($\uparrow$)</td>
<td>0.759</td>
<td>0.626</td>
<td>0.860</td>
</tr>
<tr>
<td></td>
<td>Diversity ($\downarrow$)</td>
<td>0.132</td>
<td>0.767</td>
<td>0.546</td>
</tr>
<tr>
<td></td>
<td>Distribution Sim. ($\downarrow$)</td>
<td>0.748</td>
<td>0.648</td>
<td>0.757</td>
</tr>
</tbody>
</table>

Table 8: Human evaluation results on three metrics. (The full score is 2)

Compare with the mining-based method, although REGEN costs longer time in total, we think it is worthwhile as REGEN outperforms it on all nine tasks studied in this work.

5.10 Quality Analysis of Synthetic Datasets

We provide other measurements to better evaluate the quality of the generated dataset of REGEN and baselines (Ye et al., 2022a).

Automatic Evaluations. We first measure the quality of the dataset from three perspectives: correctness, diversity, distribution similarity. The details are shown in Appendix I.1. Overall, the diversity of generated text from NLG models (Meng et al., 2022) is not satisfactory, and the correctness of text from NLG models is also not guaranteed for topic classification tasks. For the mining-based method, despite it achieves better diversity, the performances on other two metrics are worse. As a result, REGEN surpasses it on these tasks.

Human Evaluations. We also conduct human evaluations to evaluate the quality of the synthetic dataset using AG News and Sentiment datasets as two examples. For each class, we randomly sample 25 documents and ask 4 human volunteers to evaluate the dataset from three perspectives: Correctness, Informativeness and Diversity (details in Appendix I.2). The mean ratings are shown in Table 8. The average Fleiss’ Kappa (Fleiss, 1971) for correctness, informativeness and diversity are 0.53/0.57/0.58 (Moderate Agreement), respectively. Overall, the dataset curated by REGEN has the best informativeness and diversity, while has a competitive result on correctness score. These results indicate that REGEN improves over previous works for curating a better dataset to tackle the downstream

11789
tasks. Detail cases of samples from the synthetic datasets can be found at Appendix J.

6 Discussion and Conclusion

6.1 Discussion

Extending REGEN to Specific Domains. The REGEN framework is versatile and can be applied to various domains beyond our experiments. For example, it is possible to extend REGEN to zero-shot biomedical text classification (Cohan et al., 2020) using the publicly available PubMed articles as the unlabeled corpus.

Verbalizers Selection for REGEN. All the verbalizers used in this work are from the prior works (Hu et al., 2022; Schick and Schütze, 2021a) to circumvent manual prompt engineering and ensure a fair comparison. For those datasets where verbalizers are not given, we can adopt automatic verbalizer and template generation approaches (Gao et al., 2021a) to generate verbalizers for retrieving relevant documents.

Soliciting Human Feedbacks to Improve REGEN. In many cases, there may exist difficult examples where the classifier and the retrieval model do not agree with each other. To enable the model to learn on these hard examples, active learning can be adopted to solicit human annotations (Yuan et al., 2020; Yu et al., 2022a,c) or instructions (Peng et al., 2023; Zhang et al., 2022b,a) to further improve the model performance.

Collaboration with Large Language Models. There are many other potential ways to incorporate black-box large language models into REGEN beyond our experiments. For instance, large language models can be used to rerank the top retrieved documents (Ma et al., 2023) or generate augmented examples for classifiers (Møller et al., 2023). On the other hand, REGEN can be integrated into the training set synthesis for language models when the labeled dataset is inaccessible (Zhang et al., 2023). It is still an open question on how to harness large language models for dataset generation in an efficient and effective way.

6.2 Conclusion

In this paper, we propose a framework REGEN for zero-shot text classification, which incorporates dense retrieval to synthesize task-specific training sets via retrieving class-relevant documents from the generic unlabeled corpus with verbalizers. We further propose two simple while effective strategies to progressively improve the quality of the curated dataset. The effectiveness of REGEN is validated on nine benchmark datasets with an average gain of 4.3%. Further qualitative analysis justify the better quality of datasets generated by REGEN over baselines under multiple criteria.

Limitations

Our method REGEN is a general framework for zero-shot text classification. In this work, we aim to first bring in simple and intuitive way to justify the power of unsupervised dense retrieval for zero-shot learning. Effective as it is, there is still much room for improvements, including designing better objectives for pretraining $R_θ$ as well as better strategies for removing noisy training data (Lang et al., 2022; Xu et al., 2023). How to improve these components is an important line of future work.

Besides, our experiment results are all based on BERTbase sized models. Although REGEN performs on par with or better than previous dataset generation methods using giant NLG models, it remains unknown to us how the benefit of REGEN scales with more parameters for both $R_θ$ and $C_φ$.

Also, we point out that this work focuses on zero-shot text classification with task-specific verbalizers and unlabeled generic corpus, thus it can be nontrivial to adapt our framework to other tasks such as Natural Language Inference (NLI) as well as low-resource tasks where even the unlabeled generic corpus can be hard to collect. Extending REGEN to these settings will reduce the annotation burden under more challenging scenarios.

Ethics Statement

One potential risk of applying REGEN is that the generic corpus used in our experiments may contain harmful information as they were crawled from the Internet that are only filtered with some rules (Gehman et al., 2020). As a result, they may contain text exhibiting biases that are undesirable for target tasks. To alleviate this issue, we recommend the potential users to first use bias reduction and correction techniques (Schick et al., 2021) to remove biased text from the corpus to mitigate the risks of the curated dataset.

Acknowledgements

We thank anonymous reviewers for their feedback. This work was supported by NSF IIS-2008334, IIS-2106961, and CAREER IIS-2144338.
References


Ruining He and JulianMcAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Proceedings of


Yu Meng, Jiaxian Huang, Guangyuan Wang, Zihan Wang, Chao Zhang, Yu Zhang, and Jiawei Han. 2020a. Discriminative topic mining via category-name guided text embedding. In Proceedings of The Web Conference 2020, WWW ’20, page 2121–2132, New York, NY, USA. Association for Computing Machinery.

Yu Meng, Jiaxian Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. In Advances in Neural Information Processing Systems.

Yu Meng, Yunyi Zhang, Jiaxian Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020b. Text classification using label names only: A language model self-training approach. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 466–482. Springer.
Language Processing (EMNLP), pages 9006–9017, Online. Association for Computational Linguistics.


Yue Yu, Lingkai Kong, Jieyu Zhang, Rongzhi Zhang, and Chao Zhang. 2022a. AcTune: Uncertainty-based active self-training for active fine-tuning of pretrained


A Verbalizers and Templates for Datasets
The verbalizers and templates of datasets are shown in table 9.

B Corpus
We select three types of corpus, i.e. Wiki (Petroni et al., 2021), subsets of Reviews (He and McAuley, 2016) and Real News (Zellers et al., 2019) to form the corpus $D$. We manually remove documents less than 10 words as we observe that these documents do not contain informative content. The detailed information is shown in table 10.

C Baselines
We consider multiple baselines for zero-shot text classification. The details of these baselines are described as follows. We use * to denote baselines with extra resources or large language models.

Zero-shot Inference Methods These methods directly inference over the test set for prediction.

- **NSP-BERT** (Sun et al., 2022b): It uses the next sentence prediction (NSP) task to perform zero-shot learning. Specifically, it constructs prompts for each label, and use the PLM with the NSP head as the indicator.

- **Prompt** (Schick and Schütze, 2021a): It uses the original masked language modeling (MLM) objective with category-specific verbalizers to infer the true label of each sentence.

- **KNN-Prompt** (Shi et al., 2022): It improves zero-shot prompting by retrieving relevant information from an additional heterogeneous corpus, which achieves better coverage of the verbalizers.

- **KPT** (Hu et al., 2022): It uses additional knowledge bases (e.g. WordNet) to expand the label word space for verbalizers, for improving prompt-based learning.

- **GPT-3** (Brown et al., 2020): It adopts GPT-3 for zero-shot learning. We use the contextual calibration (Zhao et al., 2021) by default as it can improve the zero-shot prediction accuracy.

Weakly-supervised Learning Methods This line of methods is close to the general zero-shot learning in the sense that it does not rely on any labeled examples for classification (Shen et al., 2021; Liang et al., 2020; Zhang et al., 2021). Instead, it leverages class-specific verbalizers as well as task-specific unlabeled data as weak supervision for classification.

- **LOTClass** (Meng et al., 2020b): It first matches the label name with the corpus to find category-indicative words, then trains the model to predict their implied categories with self-training.

- **X-Class** (Wang et al., 2021): It estimates class representations by adding the most similar word to each class, then obtains the document representation with weighted average of word representations. Finally, the most confidence words are selected to fine-tune the classifier.

Note that we present the results for the two methods, but mainly for reference purposes as the setting between these approaches and our work is different.

Transfer-learning based Inference Methods

- **TE-NLI** (Yin et al., 2019): It uses the model fine-tuned on NLI tasks to perform zero-shot classification.

- **NLI-ST** (Gera et al., 2022): It uses self-training to finetune the model on additional unlabeled task-specific corpus.

We are aware that there exist some other models for generic zero-shot learning on NLP such as FLAN (Wei et al., 2022) and T0 (Sanh et al., 2022), we do not compare with them since they leverage the labeled data from some of the datasets evaluated in this work (e.g. AGNews, IMDB, according to their original paper). It is thus inappropriate to use them under the true zero-shot learning setting, since such models can have zero-shot learning unfair advantages due to access to related data during pre-training.

Dataset Generation Methods These methods generates specific datasets for zero-shot learning. Note that we use the same pretrained RoBERTa-base model as the classifier and use the same label smoothing loss for fine-tuning.

- **SuperGen** (Meng et al., 2022): It is one of the representative methods for using large natural language generation models (NLG) for zero-shot learning. It first uses the NLG model to generate training data with prompts, then selects data with highest generation probability for fine-tuning.
Table 9: The format of verbalizers and the template used for retrieval and prompting. We use the prompt formats provided in prior works (Schick and Schütze, 2021a; Hu et al., 2022). The [VERB] stands for the verbalizers. $x^a$ stands for the title (only exist in DBPedia and Yahoo) and $x^b$ stands for the body of the target document.

<table>
<thead>
<tr>
<th>Task</th>
<th>Verbalizers</th>
<th>Template used for Retrieval</th>
<th>Template used for Prompting</th>
</tr>
</thead>
<tbody>
<tr>
<td>AG News</td>
<td>politics, sports, business, technology</td>
<td>[VERB] News.</td>
<td>The category of $x^b$ is [VERB].</td>
</tr>
<tr>
<td>DBPedia</td>
<td>company, school, artist, athlete, politics, transportation, building, river/mountain/lake, village, animal, plant, album, film, book</td>
<td>[VERB]</td>
<td>$x^a$, $x^b$? The category of $x^a$ is [VERB].</td>
</tr>
<tr>
<td>Yahoo</td>
<td>society, science, health, school computer, sports, business, music, family, politics</td>
<td>[VERB]</td>
<td>$x^a$, $x^b$? The category of $x^a$ is [VERB].</td>
</tr>
<tr>
<td>NYT</td>
<td>business, politics, health, education, estate art, science, technology</td>
<td>[VERB] News.</td>
<td>The category of $x^b$ is [VERB].</td>
</tr>
<tr>
<td>Sentiment</td>
<td>great, bad</td>
<td>It was a [VERB] movie.</td>
<td>It was a [VERB] movie. $x^b$.</td>
</tr>
</tbody>
</table>

Table 10: The information about the general corpus $\mathcal{D}$ used in this study.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Size</th>
<th>Size after Pre-processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiki (Petroni et al., 2021)</td>
<td>6M</td>
<td>6M</td>
</tr>
<tr>
<td>News (Zellers et al., 2019)</td>
<td>11.9M</td>
<td>6M</td>
</tr>
<tr>
<td>Reviews (He and McAuley, 2016)</td>
<td>24.0M</td>
<td>4M</td>
</tr>
</tbody>
</table>

- **Mining** (van de Kar et al., 2022): It uses regular expressions with category-related keywords to mine samples (the next sentences of the matched text) from the corpus to generate training data. Then, it uses the zero-shot prompting to filter the noisy sample and fine-tune another classification model on the filtered dataset. For fair comparison, we use the same corpus $\mathcal{D}$, prompt format as ours for zero-shot learning, note that these often result in better performance.

The comparison of REGEN with other methods within this category (e.g. (Ye et al., 2022a,b)) is shown in Appendix F.

### D Implementation Details

#### D.1 Implementation Details for Baselines

For **zero-shot inference** methods, we directly use the numbers from the original papers if available, and reimplement Dataless and Prompt on our own. From our experiments, we observe that the numbers reported in van de Kar et al. (2022) is much lower than our reimplemented prompt-based zero-shot learning results, for reasons unknown to us.

For **transfer-learning based zero-shot inference** methods, we use the same verbalizer as REGEN and the prompt template provided from the authors for inference with the released pretrained models.

For weakly-supervised learning and zero-shot dataset generation methods, we use the code released by the authors with the optimal hyperparameters reported in the corresponding paper if available. As the code for (van de Kar et al., 2022) is not publicly available, we reimplement this method based on the information from the paper. If fine-tuning is involved, we use the same pretrained RoBERTa-base as the classifier $C_\phi$ with the label smoothing strategy for fair comparison.

#### D.2 Implementation Details for REGEN

Table 11 lists the hyperparameters used for REGEN. Note that we keep them same across all tasks without any further tuning. Under the zero-shot learning setting, there is no validation set available. For each task, we follow (Ye et al., 2022b) to use a portion (e.g., 10%) of the pseudo dataset as the validation set for model selection. If the total number of the training data for a specific category exceeds 3000, we randomly sample a subset with 3000 samples for that category.
Table 11: Hyperparameters on different tasks (they are kept same for all tasks). \( \text{lr}_f \): Learning rate for fine-tuning; \( \text{lr}_{cl} \): Learning rate for unsupervised contrastive learning; \( \text{bsz}_f \): Batch size; \( \text{bsz}_{cl} \): Batch size for unsupervised contrastive learning; \( |\tilde{T}| \): Maximum number of selected training data per class after the final retrieval round; \( E_1 \): Number of epochs for fine-tuning; \( E_2 \): Number of epochs for contrastive learning; \( T \): Number of retrieval rounds, \( \alpha \): Parameter for label smoothing ; \( \tau \): Temperature parameter for contrastive learning; \((k_1, k_2, k_3)\): Parameter \( k \) used in ANN in each round.

<table>
<thead>
<tr>
<th>Task</th>
<th>Template ID</th>
<th>Verbalizers</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGNews</td>
<td>#0 (Original)</td>
<td>politics, sports, business, technology</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>world, football, stock, science</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>international, basketball, financial, research</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>global, tennis, profit, chemical</td>
</tr>
<tr>
<td>Sentiment</td>
<td>#0 (Original)</td>
<td>great, bad</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>good, awful</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>awesome, terrible</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>incredible, horrible</td>
</tr>
</tbody>
</table>

Table 12: Different verbaliers used for expriments in section 5.7.

D.3 Number of Parameters in REGEN
The retrieval model \( R_\theta \) uses BERT-base-uncased as the backbone with 110M parameters, and the classification model \( C_\phi \) uses RoBERTa-base as the backbone with 125M parameters.

D.4 Computation Environment
All experiments are conducted on CPU: Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz and GPU: NVIDIA GeForce RTX A5000 GPUs using python 3.8 and Pytorch 1.10.

E Additional Information on Experiments Setups

E.1 Setup for Fine-tuning \( C_\phi \) with Few Labeled Examples
Under the few-shot setting, we follow (Meng et al., 2022) to split the data into two parts: half the data as training set, and the remaining as the validation set. When a few labeled samples are available, we first fine-tune the classifier \( C_\phi \) on the few-shot training set (denoted as \( C_\phi^{\text{init}} \)), and use \( C_\phi^{\text{init}} \) to remove the noisy instances with the method in Eq. 7 for both our method and baselines. Then, we continue fine-tuning the classifier on the generated data.

E.2 Setup for Zero-shot Learning with Different Verbalizers
We list the set of verbalizers used for section 5.7 in table 12.

E.3 Setup for Large Language Models for Verbalizer Expansion
For verbalizer expansion, we use GPT-4 (OpenAI, 2023) as the LLM backbone, and the prompt format is shown in the followings:

Suppose you are asked to perform text classification with the following labels. Can you generate 10 relevant keywords for each of the categories?

By inputting the verbalizers of each class into the chatbox, the LLM can output a series of keywords to enrich the verbalizer. After obtaining the keywords, we manually remove keywords that occur in more than one category, and the remaining keywords will be used for retrieval.

F Comparison with Recent Baselines
We provide additional empirical studies to compare REGEN with some recent works. As (Ye et al., 2022a,b) use a smaller PLM, namely DistillBERT (Sanh et al., 2019) for their experiments, we use the same DistillBERT encoder to finetune our model and several baselines (e.g. Mining (van de Kar et al., 2022) and SuperGen (Meng et al., 2022)). The result is shown in table 13.

Overall, we observe that REGEN outperforms most of these baselines with DistillBERT as the classifier. It achieves competitive performance with ProGen, which relies on several additional techniques including influence estimation, multi-round in-context feedback from a billion-scale language model, and noise-robust loss functions. Note that these techniques are orthogonal to our method, and can be potentially integrated with REGEN for better performance.
<table>
<thead>
<tr>
<th>Method/Dataset</th>
<th>IMDB</th>
<th>SST-2</th>
<th>Rotten Tomato</th>
<th>Elec</th>
<th>Yelp</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prompting*</td>
<td>77.31</td>
<td>82.63</td>
<td>78.66</td>
<td>78.03</td>
<td>80.30</td>
<td>79.39</td>
</tr>
<tr>
<td>ZeroGen* (Ye et al., 2022b)</td>
<td>80.41</td>
<td>82.77</td>
<td>78.36</td>
<td>85.35</td>
<td>87.84</td>
<td>83.29</td>
</tr>
<tr>
<td>ProGen* (Ye et al., 2022a)</td>
<td>84.12</td>
<td>87.20</td>
<td>82.86</td>
<td>89.00</td>
<td>89.39</td>
<td>86.51</td>
</tr>
<tr>
<td>SuperGen (Meng et al., 2022)</td>
<td>84.58</td>
<td>86.70</td>
<td>79.08</td>
<td>90.58</td>
<td>89.98</td>
<td>86.18</td>
</tr>
<tr>
<td>Mining (van de Kar et al., 2022)</td>
<td>77.36</td>
<td>80.73</td>
<td>76.73</td>
<td>85.87</td>
<td>90.36</td>
<td>82.21</td>
</tr>
<tr>
<td>REGEN</td>
<td>87.84</td>
<td>85.32</td>
<td>81.42</td>
<td>89.83</td>
<td>89.00</td>
<td>86.68</td>
</tr>
</tbody>
</table>

Table 13: Results with recent baselines using DistillBERT (Sanh et al., 2019) as \( C_\phi \). *: Results are copied from the previous papers (Ye et al., 2022a,b).

<table>
<thead>
<tr>
<th>Task</th>
<th>Datasets</th>
<th>Performance of REGEN</th>
<th>Fully-supervised Performance</th>
<th>( \Delta ) Performance Gap</th>
<th>Lexical Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic</td>
<td>AG News</td>
<td>85.0</td>
<td>94.6</td>
<td>10.0%</td>
<td>0.427</td>
</tr>
<tr>
<td></td>
<td>DBPedia</td>
<td>87.6</td>
<td>99.2</td>
<td>11.2%</td>
<td>0.566</td>
</tr>
<tr>
<td></td>
<td>Yahoo</td>
<td>59.4</td>
<td>76.8</td>
<td>17.4%</td>
<td>0.362</td>
</tr>
<tr>
<td></td>
<td>NYT</td>
<td>74.5</td>
<td>88.2</td>
<td>15.5%</td>
<td>0.530</td>
</tr>
<tr>
<td>Sentiment</td>
<td>IMDB</td>
<td>89.9</td>
<td>94.4</td>
<td>4.5%</td>
<td>0.497</td>
</tr>
<tr>
<td></td>
<td>MR</td>
<td>82.5</td>
<td>91.3</td>
<td>8.8%</td>
<td>0.306</td>
</tr>
<tr>
<td></td>
<td>SST-2</td>
<td>88.9</td>
<td>96.2</td>
<td>7.3%</td>
<td>0.296</td>
</tr>
<tr>
<td></td>
<td>Amazon</td>
<td>92.3</td>
<td>95.4</td>
<td>3.1%</td>
<td>0.714</td>
</tr>
<tr>
<td></td>
<td>Yelp</td>
<td>93.0</td>
<td>97.2</td>
<td>4.2%</td>
<td>0.408</td>
</tr>
</tbody>
</table>

Table 14: The detailed value for the performance gap and the lexical similarity between the task-specific corpus and the general-domain corpus \( D \).

**G More Details on Performance Gaps and Lexical Similarities**

**G.1 Calculating the Similarity between the Corpus and Target Tasks**

We use the weighted Jaccard similarity \( J(T, D) \) to measure distribution similarities between the corpus \( D \) and the target task \( T \), described as follows: Denote \( C_k \) as the frequency of word \( k \) in the corpus \( D \) and \( T_k \) for the target task \( T \) respectively. The weighted Jaccard similarity \( J(T, D) \) is defined as:

\[
J(T, D) = \frac{\sum_k \min\{C_k, T_k\}}{\sum_k \max\{C_k, T_k\}},
\]

where the sum is over all unique words \( k \) present in \( D \) and \( T \).

**G.2 The Performance Gap and Lexical Similarity for All Datasets**

The details for the performance gap as well as the lexical similarity to the general-domain corpus are shown in Table 14.

**H Additional Per-task Results**

We show the results for each task in this section. Specifically, we present the performance of REGEN and its variation of without the filtering step in Fig. 6; we present the performance of REGEN with different dense retrieval models as \( R_\theta \) in Fig. 7; we illustrate the performance under different volume of training data for REGEN and baselines in Fig. 8; we demonstrate the effect of different corpus \( D \) on the final performance in Fig. 9. Besides, in table 15 we illustrate the performance of REGEN and baselines on all sentiment analysis datasets; in table 16, the automatic evaluation results for all datasets are shown.
Figure 6: Effect of filtering, per task results.

Figure 7: Comparisons of different dense retrieval models, per task results.

Figure 8: Performance of the different amount of training data, per task results.

Figure 9: Performance of REGEN using different subsets of corpus on other sentiment classification tasks.
Table 16: Automatic evaluation results on all datasets. Note that we only generate one dataset for all sentiment analysis tasks.

I Details for Quality Analysis

I.1 Automatic Evaluation

We provide the details for automatic measurements of the dataset quality as follows.

For correctness, we first fine-tune a RoBERTa-Large model on the original dataset, and use the fine-tuned model as an oracle to evaluate the correctness of the synthetic dataset.

For diversity, we use the self-BLEU (Zhu et al., 2018), which computes the BLEU-4 score of each generated text with other generations in the dataset as references, as the metric. Note that for self-BLEU, a lower score implies higher diversity.

Besides, we use MAUVE (Pillutla et al., 2021) with the default hyperparameter settings to measure the distribution similarity. MAUVE is originally proposed for comparing the learnt distribution of a text generation model and the distribution of human-written text, and we adapt MAUVE to measure the similarity between the distribution of the synthetic dataset and the real dataset. A higher value indicates that the distribution of the synthetic dataset and the real dataset is closer, thus the quality of the synthetic dataset is higher.

I.2 Human Evaluation

Apart from the automatic evaluation, we also perform human evaluation to manually evaluate the quality of the synthetic dataset. We ask four volunteer students from our institute (approved by the ethics review board) for participation. For human evaluation, the evaluation form is listed as below.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metrics</th>
<th>Mining</th>
<th>SuperGen</th>
<th>REGEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentiment</td>
<td>Correctness (↑)</td>
<td>0.815</td>
<td>0.971</td>
<td>0.986</td>
</tr>
<tr>
<td></td>
<td>Diversity (↓)</td>
<td>0.144</td>
<td>0.915</td>
<td>0.361</td>
</tr>
<tr>
<td></td>
<td>Distribution Sim. (↑)</td>
<td>0.856</td>
<td>0.803</td>
<td>0.865</td>
</tr>
<tr>
<td>AG News</td>
<td>Correctness (↑)</td>
<td>0.746</td>
<td>0.649</td>
<td>0.805</td>
</tr>
<tr>
<td></td>
<td>Diversity (↓)</td>
<td>0.117</td>
<td>0.818</td>
<td>0.330</td>
</tr>
<tr>
<td></td>
<td>Distribution Sim. (↑)</td>
<td>0.799</td>
<td>0.687</td>
<td>0.686</td>
</tr>
<tr>
<td>DBPedia</td>
<td>Correctness (↑)</td>
<td>0.791</td>
<td>0.516</td>
<td>0.909</td>
</tr>
<tr>
<td></td>
<td>Diversity (↓)</td>
<td>0.223</td>
<td>0.765</td>
<td>0.377</td>
</tr>
<tr>
<td></td>
<td>Distribution Sim. (↑)</td>
<td>0.874</td>
<td>0.662</td>
<td>0.920</td>
</tr>
<tr>
<td>NYT</td>
<td>Correctness (↑)</td>
<td>0.730</td>
<td>0.811</td>
<td>0.893</td>
</tr>
<tr>
<td></td>
<td>Diversity (↓)</td>
<td>0.100</td>
<td>0.717</td>
<td>0.342</td>
</tr>
<tr>
<td></td>
<td>Distribution Sim. (↑)</td>
<td>0.511</td>
<td>0.643</td>
<td>0.622</td>
</tr>
<tr>
<td>Yahoo</td>
<td>Correctness (↑)</td>
<td>0.771</td>
<td>0.518</td>
<td>0.832</td>
</tr>
<tr>
<td></td>
<td>Diversity (↓)</td>
<td>0.089</td>
<td>0.768</td>
<td>0.335</td>
</tr>
<tr>
<td></td>
<td>Distribution Sim. (↑)</td>
<td>0.810</td>
<td>0.602</td>
<td>0.797</td>
</tr>
</tbody>
</table>

- **Correctness**: Whether the text is relevant to the corresponding label?
  - 2: Accurate: The content is accurate for the label.
  - 1: Related: The content is related but not accurate for the label.
  - 0: Not relevant: The content is not relevant to the label.

- **Informativeness**: Whether the text is fluent and similar to human-generated text?
  - 2: Very Informative: The text is very informative and similar to human generated text.
  - 1: Partially Informative: The text is partially informative and somewhat close to human generated text.
  - 0: Not Informative: The text is not fluent/informative at all.

J Case Studies

We present some examples of the curated dataset in the table 17 and 18. Note that filtered means the data is first retrieved by $R_\theta$ but is later identified as incorrect sample by the classifier. Overall, we observe that the dataset of SuperGen contains similar sentences across samples (e.g. a great example of the kind of movie for sentiment analysis datasets), and the mining-based approach often contains documents that are less informative (e.g. World famous hapuna beach is just minutes away for topic classification). In contrast, REGEN goes beyond the exact keyword matching and can retrieve diverse and informative documents. Moreover, the noisy samples can be filtered through self-consistency regularization. These cases corroborate the result on both automatic and human evaluation, and justify the higher quality of the dataset curated by us.

We also demonstrate the retrieved examples over different rounds in table 19 and 20. Note that examples shown in the 2nd and 3rd round are retrieved...
The film is a great example of the kind of movie that you can watch over and over.

The movie was very good and it had a lot of action in it. I would recommend this to anyone who likes action.

This film is a great example of the kind of movie that you can watch with your kids and not have to worry about anything inappropriate.

The film was a total waste of time. I would not recommend this movie to anyone.

This is a must see film for all ages I would have given this film 10 stars if they would have let me. This is one of those films that somehow got overlooked in the theaters.

Worst movie ever A good example of what is wrong with Hollywood today. I have never looked at my watch more than I did during this movie.

OK, this cd makes me sad.

Great I bought this toy for my son’s 3rd birthday and only after 2 months he now sings the alphabet song all the time. It is a great education toy and also very durable.

After seeing the movie "12 Years A Slave," I wanted to read the book. The experience of watching the movie drew me into the story of Solomon Northup’s life.

This is a must see film for all ages I would have given this film 10 stars if they would have let me. This is one of those films that somehow got overlooked in the theaters.

Excellent but still not Perfect. Don’t take my title or rating the wrong way. My experience with the first 2 Harry Potter Movies have been extremely good but in the 2nd movie, the Chamber of Secrets, A lot of parts were taken out...

OK, this cd makes me sad.

The film was a total waste of time. I would not recommend this movie to anyone.

This film is a waste of time. It has no plot and the acting was terrible. I would not recommend this movie to anyone.

This movie is not worth the time or money to watch. It was a waste of my time.

The dvd arrived very quick.

I can’t stop playing them right now.

I cannot use it on the hardwood floors because I am afraid water might get down under them (they are very old but have been refinished.).

The plastic handle is not able to be taken apart so I don’t know where the leak was exactly coming from.

Don’t know this for sure, but it seems likely.

Table 17: Example retrieved texts of ReGen and two baselines on synthetic dataset for sentiment analysis.

<table>
<thead>
<tr>
<th>Method</th>
<th>Label</th>
<th>Generated Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>SuperGen</td>
<td>positive</td>
<td>The film is a great example of the kind of movie that you can watch over and over.</td>
</tr>
<tr>
<td>SuperGen</td>
<td>positive</td>
<td>The movie was very good and it had a lot of action in it. I would recommend this to anyone who likes action.</td>
</tr>
<tr>
<td>SuperGen</td>
<td>positive</td>
<td>This film is a great example of the kind of movie that you can watch with your kids and not have to worry about anything inappropriate.</td>
</tr>
<tr>
<td>SuperGen</td>
<td>negative</td>
<td>The film was a total waste of time. I would not recommend this movie to anyone.</td>
</tr>
<tr>
<td>SuperGen</td>
<td>negative</td>
<td>This film is a waste of time. It has no plot and the acting was terrible. I would not recommend this movie to anyone.</td>
</tr>
<tr>
<td>SuperGen</td>
<td>negative</td>
<td>This movie is not worth the time or money to watch. It was a waste of my time.</td>
</tr>
<tr>
<td>Mining</td>
<td>positive</td>
<td>The dvd arrived very quick.</td>
</tr>
<tr>
<td>Mining</td>
<td>positive</td>
<td>I can’t stop playing them right now.</td>
</tr>
<tr>
<td>Mining</td>
<td>positive</td>
<td>I cannot use it on the hardwood floors because I am afraid water might get down under them (they are very old but have been refinished.).</td>
</tr>
<tr>
<td>Mining</td>
<td>negative</td>
<td>The plastic handle is not able to be taken apart so I don’t know where the leak was exactly coming from.</td>
</tr>
<tr>
<td>Mining</td>
<td>negative</td>
<td>Don’t know this for sure, but it seems likely.</td>
</tr>
</tbody>
</table>

Table 18: Example retrieved texts of ReGen and two baselines on the synthetic dataset for AG News.

<table>
<thead>
<tr>
<th>Method</th>
<th>Label</th>
<th>Generated Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>AG News</td>
<td>politics</td>
<td>The opinions expressed in this commentary are solely those of John Avlon..</td>
</tr>
<tr>
<td>AG News</td>
<td>politics</td>
<td>At the same time, we should not let our good fortune make us callous to the effect of suffering on most of the world population.</td>
</tr>
<tr>
<td>AG News</td>
<td>politics</td>
<td>TL;DR Correction of Sept 30 article on Pres Bush’s visit to New York City, which misstated his role in campaign finance reform legislation that was signed into law by Gov George Pataki.</td>
</tr>
<tr>
<td>AG News</td>
<td>technology</td>
<td>TL;DR The National Science Foundation awarded $32 million to the University of California, Berkeley, for... Times Magazine publishes its annual list of the 100 most influential people in science, technology, engineering or math.</td>
</tr>
<tr>
<td>AG News</td>
<td>business</td>
<td>THE HAGUE, Netherlands, March 14, 2019 /PRNewswire/ – Royal Dutch Shell plc RDS.A, +0.35% RDS.B, +0.19% filed its Annual Report on Form 20-F for the year ended December 31, 2018, with the U.S. Securities and Exchange Commission.</td>
</tr>
<tr>
<td>AG News</td>
<td>sports</td>
<td>Manchester City’s quest for four trophies continued with a 5-0 thrashing of Burnley to march into the FA Cup fifth round as League One Shrewsbury narrowly missed out on shocking Wolves in a 2-2 draw on Saturday.</td>
</tr>
<tr>
<td>AG News</td>
<td>sports</td>
<td>Tom Brady and Bill Belichick likely will go down as the greatest quarterback/coach combo in NFL history, ... Super Bowl together with a thrilling 34-28 overtime victory against the Atlanta Falcons in Super Bowl LI on Sunday night.</td>
</tr>
<tr>
<td>AG News</td>
<td>politics</td>
<td>Police in Bolivia have rebelled against the government, abandoning their posts and marching through the streets along with protesters. It’s a sign of growing anger over alleged voter fraud in last month’s election. Protests since the poll have resulted in three deaths.</td>
</tr>
<tr>
<td>AG News</td>
<td>technology</td>
<td>TL;DR Police in association with a bangalore-based start-up has come up with a technology which can detect unauthorized drones.</td>
</tr>
</tbody>
</table>

Table 19: Example retrieved texts of ReGen and two baselines on the synthetic dataset for sentiment analysis.
**Table 19:** Example retrieved texts of REGEN over three rounds for sentiment datasets.

<table>
<thead>
<tr>
<th>Round</th>
<th>Label</th>
<th>Generated Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>positive</td>
<td>“Deceptions” was one of the best films I have seen in a long time. Stefanie Powers was excellent as Sabrina and Samantha. The rest of the cast was also very good.</td>
</tr>
<tr>
<td></td>
<td>negative</td>
<td>I honestly have no idea what to say about this movie. It literally left me speechless…in a very, very not-good way.</td>
</tr>
<tr>
<td>2</td>
<td>positive</td>
<td>I saw the film last weekend and enjoyed it. From the point of view of movie craftsmanship, it’s hard to go wrong with the talent combination of Steven Spielberg, Meryl Streep, Tom Hanks, and John Williams.</td>
</tr>
<tr>
<td></td>
<td>negative</td>
<td>To be frank, it is a really bad movie. The cheap symbolism would make a junior high English teacher blush (including the title), and the lopsided view of racism in America was painfully and repeatedly portrayed.</td>
</tr>
<tr>
<td>3</td>
<td>positive</td>
<td>&quot;Letting Go,&quot; with Sharon Gless and John Ritter, was a warm, funny and dramatic movie. I loved it. It was a fresh and wonderful romance.</td>
</tr>
<tr>
<td></td>
<td>negative</td>
<td>First of all, I would like to say that I think the movie did an excellent job of following the events in the book. But they did a pretty bad job of leaving some crucial parts out of the movie. In the book, you get a pretty strong sense of the bond and relationship between the characters. In the movie, you don’t really see that bond at all.</td>
</tr>
</tbody>
</table>

**Table 20:** Example retrieved texts of REGEN over three rounds for AG News dataset.

<table>
<thead>
<tr>
<th>Round</th>
<th>Label</th>
<th>Generated Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>politics</td>
<td>The UN voiced hope Monday that a meeting this week of a committee tasked with amending Syria’s constitution can open the door to a broader political process for the war-ravaged country.</td>
</tr>
<tr>
<td></td>
<td>sports</td>
<td>LaLiga may boast football superpowers Real Madrid and Barcelona but the league is keen to help other Spanish sports succeed too.</td>
</tr>
<tr>
<td></td>
<td>business</td>
<td>Corporate America is slowly starting to give cash back to investors with dividends and buybacks. Companies are also spending cash on mergers.</td>
</tr>
<tr>
<td></td>
<td>technology</td>
<td>Google said on Wednesday it had achieved a breakthrough in research, by solving a complex problem in minutes with a so-called quantum computer that would take today’s most powerful supercomputer thousands of years to crack.</td>
</tr>
<tr>
<td>2</td>
<td>politics</td>
<td>The death toll in Eastern Ghouta stands at nearly 500, and it remains unclear how the sustained bombing campaign in the region will stop—despite a UN vote.</td>
</tr>
<tr>
<td></td>
<td>sports</td>
<td>Barcelona continued their quest to win La Liga with a comfortable 3-0 victory over Leganes yesterday. Luis Suarez ended his goal drought with a brilliant brace before summer signing Paulinho got on the scoresheet late on.</td>
</tr>
<tr>
<td></td>
<td>business</td>
<td>For many American companies today it is almost as is the recession never happened as executive incomes rise above pre-recession levels. According to Standard &amp; Poor’s 500 the average income of an executive in 2010 was $9 million. That is 24 percent higher than it was the year prior.</td>
</tr>
<tr>
<td></td>
<td>technology</td>
<td>Scientists claimed Wednesday to have achieved a near-mythical state of computing in which a new generation of machine vastly outperforms the world’s fastest super-computer, known as “quantum supremacy”</td>
</tr>
<tr>
<td>3</td>
<td>politics</td>
<td>The UN’s ceasefire in Syria’s rebel-held enclave of Eastern Ghouta was cast into doubt less than 24 hours after the Security Council voted to uphold it, as residents woke to regime airstrikes and Iran vowed to carry on fighting in areas it deems held by terrorists.</td>
</tr>
<tr>
<td></td>
<td>sports</td>
<td>Eden Hazard exploded into life and Karim Benzema continued his brilliant scoring run as Real Madrid delivered another goalfest on Saturday in a 4-0 demolition of Eibar.</td>
</tr>
<tr>
<td></td>
<td>business</td>
<td>Wall Street’s eternally optimistic forecasters are expecting corporate profit growth to surge by the middle of next year views that are about to collide with reality as hundreds of companies report financial results and update investors on their prospects.</td>
</tr>
<tr>
<td></td>
<td>technology</td>
<td>From ending the opioid epidemic to making fusion power possible, ‘Summit’ may help researchers meet all sorts of goals. A $200-million, water-cooled monster that covers an area the size of two tennis courts, the computer, dubbed “Summit,” has been clocked at handling 200 quadrillion calculations a second.</td>
</tr>
</tbody>
</table>
ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
  Limitation section

- A2. Did you discuss any potential risks of your work?
  Ethics Statement section

- A3. Do the abstract and introduction summarize the paper’s main claims?
  Abstract, Section 1

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

B ✔️ Did you use or create scientific artifacts?

Yes Section 5

- ✔️ B1. Did you cite the creators of artifacts you used?
  Section 5.1

- □ 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)?
  Not applicable. Left blank.

- ✔️ 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?
  We use public datasets available online, which have also been widely used by other studies.

- ✔️ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
  Section 5.1

- ✔️ 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.
  Appendix A

C ✔️ Did you run computational experiments?

Section 5

- ✔️ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
  Appendix D.3, D.4

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?

*Appendix D.2, table 11. We do not search hyperparameters and use _one_ hyperparameter set for all tasks.*

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?

*Yes, section 5.2 (table 1).*

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.)?

*Appendix I.1*

**D**

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

*Section 5.10*

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.?

*Appendix I.2*

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)?

*Appendix I.2*

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?

*Appendix I.2*

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.*