CELDA: Leveraging Black-box Language Model as Enhanced Classifier without Labels

Hyunsoo Cho[◇], Youna Kim[◇], Sang-goo Lee[◇♣]
[◇]Seoul National University, [♣]IntelliSys
{johyunsoo, anna9812, sglee}@europa.snu.ac.kr

Abstract

Utilizing language models (LMs) without internal access is becoming an attractive paradigm in the field of NLP as many cutting-edge LMs are released through APIs and boast a massive scale. The de-facto method in this type of *black-box* scenario is known as *prompting*, which has shown progressive performance enhancements in situations where data labels are scarce or unavailable. Despite their efficacy, they still fall short in comparison to fully supervised counterparts and are generally brittle to slight modifications. In this paper, we propose Clustering-Enhanced Linear Discriminative Analysis (CELDA), a novel approach that improves the text classification accuracy with a very weak-supervision signal (i.e., name of the labels). Our framework draws a precise decision boundary without accessing weights or gradients of the LM model or data labels. The core ideas of CELDA are twofold: (1) extracting a refined pseudo-labeled dataset from an unlabeled dataset, and (2) training a lightweight and robust model on the top of LM, which learns an accurate decision boundary from an extracted noisy dataset. Throughout in-depth investigations on various datasets, we demonstrated that CELDA reaches new state-of-the-art in weakly-supervised text classification and narrows the gap with a fully-supervised model. Additionally, our proposed methodology can be applied universally to any LM and has the potential to scale to larger models, making it a more viable option for utilizing large LMs.

1 Introduction

Large-scale language models (LMs) have been a driving force behind a series of breakthroughs in the machine-learning community. Despite their preeminence in wide applications, large LMs are often costly or infeasible to fine-tune as many distributed large models, such as GPT-3 (Brown et al., 2020), are provided in a *black-box* manner, which only allows limited access through commercial APIs.

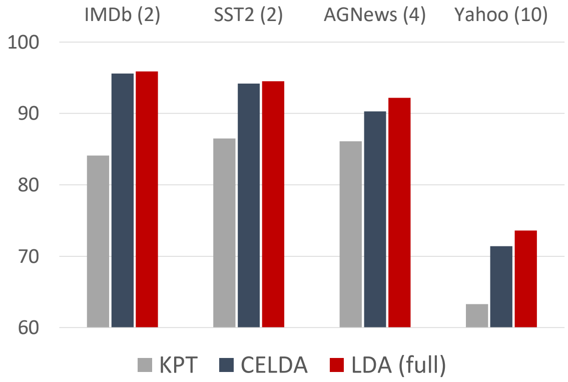


Figure 1: Text classification accuracy on 4 benchmark datasets on T5 (11B). CELDA significantly improves the performance without model adaptation or labeled data closing the gap with fully labeled methods.

To circumvent the explicit adaptation of the models, many recent research leverage *prompting*, a training-free approach that elicits desired predictions from LMs by curating input into a more conceivable form. By doing so, prompting has shown remarkable improvements in data-scarce scenarios (i.e., zero-shot, few-shot), reminiscing the potential of large LMs as a universal, off-the-shelf solution for diverse tasks. Yet, it is still premature as their performance lags far behind the fine-tuned model and displays fragility to negligible changes (Lu et al., 2022; Perez et al., 2021).

In this paper, we aim to bridge this gap by utilizing an unlabeled dataset without adapting the model and propose Clustering-Enhanced Linear Discriminative Analysis (CELDA) that maximizes the potential of *black-box* language models further. CELDA enhances text classification performance by training a lightweight module stacked on the top of LM. The improvement is a result of attaining the following two key objectives: (1) composing a highly reliable pseudo-labeled dataset with *black-box* LM. (2) training a compact but robust model with the previously composed dataset

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4364–4379
July 9-14, 2023 ©2023 Association for Computational Linguistics

(pseudo-labels). We accomplish the first goal by sorting out some uncertain data points via clustering the features from LM. We draw inspiration from recent findings (Aharoni and Goldberg, 2020; Cho et al., 2023) that LM effectively groups semantically similar sentences into clusters. Furthermore, we achieve the latter objective by employing Linear Discriminative Analysis (LDA) which is efficient in terms of parameter requirements and robust against spurious inputs (Murphy, 2022). The mentioned characteristics of LDA have strong compatibility with our training dataset, considering the presence of noise and its reduced scale.

To verify our method, we compare with recently proposed state-of-the-art methods on 8 different text classification datasets, spanning from binary to multi-class tasks (maximum 150 classes). Moreover, we also report the performance of baseline LDA when the data labels are fully available, serving as the upper bound performance. As illustrated in Figure 1, our method significantly outperforms the precedently proposed prompting-based zero-shot learning (ZSL) method (Hu et al., 2021) and closes the gap between fully labeled methods.

In summary, our contributions are as follows:
(1) We propose dubbed CeLDA, a novel weakly-supervised learning framework for black-box language models that consistently outperforms other competing methods and close the gap between fully fine-tuned model.
(2) CeLDA has the potential to scale to larger models, whereas performance often saturates in ZSL methods, and existing weakly-supervised learning methods often require additional model tuning.
(3) CeLDA proves to be a highly effective active learner, capable of tackling even the most challenging tasks previously faced by ZSL or WSL methods with minimal human-in-loop labeling.

2 Related Work

Zero-shot learning aims to identify the class label of the input sentence by relying on weak supervision information from metadata, such as the description of the task and the name of the class labels. ZSL methods inference test input instantly without any dataset or explicit training. Specifically, most ZSL works (Holtzman et al., 2021; Zhao et al., 2021; Min et al., 2022; Schick and Schütze, 2021) utilize the likelihood of manually designed verbalizers relying on the language model's capability to predict the probability of the `[MASK]` word (bi-directional models) or next token (auto-regressive models).

Additionally, most recent research in ZSL utilizes external knowledge bases or corpus (Lyu et al., 2022; Shi et al., 2022; Hong et al., 2022). Lyu et al. (2022) retrieves semantically similar samples from the additional corpus and labels them randomly. Shi et al. (2022) employs automatically expanded fuzzy verbalizers to converge the mapping between the verbalizer tokens and the class labels. Hong et al. (2022) additionally uses semantically similar sentences from supplementary corpora to compensate for the poorly described labels.

Weakly-supervised learning approaches, unlike ZSL methods, generally require an unlabeled dataset relevant to the target task and re-train the backbone model. Generally, most WSL studies (Meng et al., 2020; Wang et al., 2021; Zhang et al., 2021; Fei et al., 2022b) utilize keywords from metadata (e.g., name of each class) to annotate unlabeled datasets and iteratively re-train models with the pseudo-labeled dataset.

Namely, X-class (Wang et al., 2021) incrementally adds several similar words to each class until in-consistency arises and utilizes the weighted average of contextualized word representations to retrieve the most confident documents from each cluster to train a text classifier. SimPTC (Fei et al., 2022b) trains a Gaussian Mixture Model (GMM) on top of the LM's representations, similar to our approach in that it utilizes Gaussian distribution to fit the model. While effective, most WSL methods are challenging to utilize in *black-box* scenarios as they often require direct model tuning or are tailored for a particular language model.

Black-box tuning is a research field aiming to maximize downstream task performance without accessing weights or gradients of the model, which has diverse potential and benefits. Black-box tuned models can process *mixed-task* input batch with a single model as it circumvents the explicit model retraining phase. Furthermore, it can leverage some commercial models available only through APIs (Brown et al., 2020; Sun et al., 2021) or even when models are too large (Zhang et al., 2022; Scao et al., 2022) to optimize directly.

In this scenario, the prevailing paradigms are: (1) manipulating the input text or (2) training a lightweight model on top of the final representations from the model. Specifically, the most recent work utilizes the language model's ability to learn in-context (Brown et al., 2020) and tailors the input through appending templates or few-shot samples (i.e., demonstrations) to the original inputs. Additionally, some studies (Diao et al., 2022; Sun et al., 2022) attempt to find the optimal prompt for the task without explicitly calculating the gradient.

3 Preliminary

3.1 Scenario & Problem Formulation

Our research aims to improve text classification without using dataset labels and accessing LM's parameters or gradients. Namely, LM only serves as a fixed text encoder function $\texttt{LM}(\cdot)$, which outputs an n-dimensional continuous latent representation $\boldsymbol{h} \in \mathcal{H} = [0, 1]^n$ from input $\boldsymbol{x}$: $\boldsymbol{h} = \texttt{LM}(\boldsymbol{x})$. And let the label space be $\mathcal{Y} = \{0, 1, \cdots, |\mathcal{Y}|\}$, where $|\mathcal{Y}|$ is total cardinality of label space. Then, our goal is to build a classifier $f_{cls} : \mathcal{H} \mapsto \mathcal{Y}$ which maps encoded features to proper classes with an unlabeled dataset $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^{k}$ and weak supervision signal (i.e., natural language name for each class).

3.2 Prompt-tuning

Prompt-tuning projects an input sequence x into a label space $\mathcal{Y}$ via LM's capability to fill the `[MASK]` token. For instance, suppose we have to classify the binary sentiment (label 0: `positive`, label 1: `negative`) of the sentence. Given input sentence $\boldsymbol{x} =$ A great movie., we first transform the $\boldsymbol{x}$ into a cloze question through *template*:

$$\boldsymbol{x}^{\mathrm{t}} = \texttt{A [Mask] review.}\ \text{A great movie.}$$

Then we feed *templified* input $\boldsymbol{x}^{\mathrm{t}}$ to LM and extract the likelihood of the `[MASK]` token. To convert the probability of extracted token into a label probability, we employ a *verbalizer* $\mathcal{V}$, a few selected tokens from the whole vocabulary, and map them into corresponding label space $\mathcal{V} \mapsto \mathcal{Y}$. Specifically from the previous example, we can design a verbalizer utilizing a weak-supervision signal (i.e., name of each class) as follows:

$$\mathcal{V}_0 = \{\texttt{positive}\}, \mathcal{V}_1 = \{\texttt{negative}\}. \quad (1)$$

Note that some recent studies have expanded the verbalizer words in Eq. 1 to multiple words by utilizing extra knowledge, such as ConceptNet (Speer et al., 2017) or WordNet (Miller, 1995), to make more accurate predictions. Then, for input $\boldsymbol{x}^{\mathrm{t}}$, we measure the probability distribution of the label $P(y|\boldsymbol{x}^{\mathrm{t}})$ utilizing verbalizer $\cup_{y\in\mathcal{Y}}\mathcal{V}_y = V$:

$$P(y|\boldsymbol{x}^{\mathrm{t}}) = \frac{\sum_{w\in V_y} \texttt{LM}(\texttt{[MASK]} = w|\boldsymbol{x}^{\mathrm{t}})}{\sum_{\mathcal{V}_i\in\mathcal{V}} \sum_{v\in V_i} \texttt{LM}(\texttt{[MASK]} = v|\boldsymbol{x}^{\mathrm{t}})} \quad (2)$$

Finally, we compare the probability of $p(y = 0)$ and $p(y = 1)$ in the `[Mask]` token and annotates input $\boldsymbol{x}$ with label 0.

3.3 Linear Discriminative Analysis

LDA belongs to the generative classifier family which estimates the class probability of the input indirectly. Unlike discriminative classifier, which directly models the class posterior $p(y = c|\boldsymbol{x})$, LDA predicts $\boldsymbol{x}$ via bayes rule:

$$p(y = c|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y = c)p(y = c)}{\Sigma_{c' \in\mathcal{Y}} p(\boldsymbol{x}|y = c')p(y = c')}.$$

LDA assumes the class conditional densities follow multivariate Gaussian distribution with tied covariance over classes:

$$p(\boldsymbol{x}|y = c) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_c, \Sigma)$$

Then the corresponding label probability (posterior) has the following form:

$$\begin{aligned} p(y = c|\boldsymbol{x}) &\propto \pi_c\, \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_c, \Sigma) \\ &\propto \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_c, \Sigma). \end{aligned}$$

The prior distribution $\pi_c = p(y = c)$ can be ignored as it is independent to c.

Train: To fit the LDA model, we have to estimate the Normal distribution of the target space $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ from the available dataset $\mathcal{D}$. Particularly, LDA estimates the distribution by employing MLE, which consists of empirical class mean $\hat{\boldsymbol{\mu}}_c$ and tied covariance $\hat{\Sigma}$.

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{|\mathcal{D}_c|} \sum_{n\in\mathcal{D}_c} \boldsymbol{x}_n,$$

$$\hat{\Sigma} = \frac{1}{|\mathcal{D}|} \sum_{c=1}^{c\in\mathcal{Y}} \sum_{n\in\mathcal{D}_c} (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_c)(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_c)^{\mathsf{T}}.$$

Inference: After training the trainable parameters, we can compute the probability of the class label through Mahalanobis distance d_{mah} which measures the distance between the data point $\boldsymbol{x}$ and the estimated distribution $\mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\Sigma})$:

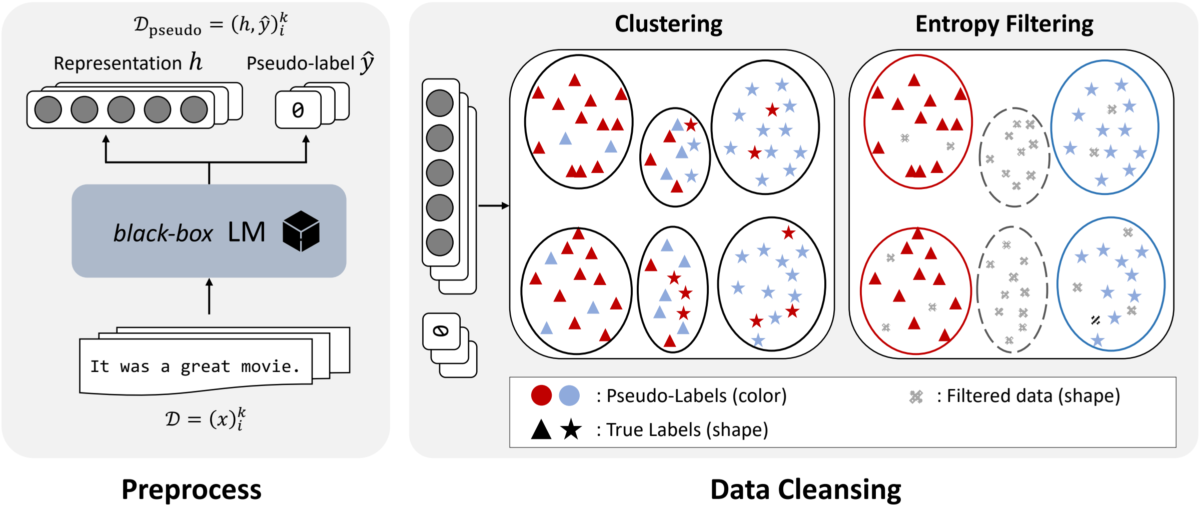


Figure 2: Illustration of § 4.1, 4.2 stage in CeLDA. We (1) extract pseudo label and latent feature pair from unlabeled data, and (2) discard some uncertain data points.

$$\begin{aligned}\hat{y} = M(\boldsymbol{x}) &= \arg\max_c \log p(y = c|\boldsymbol{x}) \\ &= \arg\max_c \left[\log \pi_c - \frac{1}{2} d_{\text{mah}}(\boldsymbol{x}, \boldsymbol{\mu}_c; \Sigma) + \mathrm{C}\right] \\ &= \arg\min_c d_{\text{mah}}(\boldsymbol{x}, \boldsymbol{\mu}_c; \Sigma),\end{aligned}$$

where M refers to a trained LDA model and $d_{\text{mah}}(\boldsymbol{x}, \boldsymbol{\mu}_c; \Sigma) = (\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_c)^{\intercal}\Sigma^{-1}(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}}_c)$.

4 CeLDA

We introduce CeLDA, Clustering-enhanced Linear Discriminative Analysis which enhances the text classification ability of *black-box* LMs without labels. CeLDA first (1) consists pseudo labeled dataset by passing an unlabeled dataset to $\texttt{LM}(\cdot)$ and extracts high-quality representation and pseudo-label pair as introduced in § 3.2. Then, we (2) refine the pseudo labeled dataset into a small subset dataset dubbed a **certain dataset** by leveraging clustering and the concept of entropy. (Figure 2 illustrates the (1), (2) stage of CeLDA training.) Finally, (3) we recursively train a third-party LDA module stacked on LM with the **certain dataset**, which learns a precise decision boundary from the noisy dataset.

4.1 Pre-processing

CeLDA first passes the unlabeled dataset $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^k$ to LM to extract the information required to train LDA. Specifically, we pass *templified* sentence $\boldsymbol{x}^{\text{t}}$ to $\texttt{LM}(\cdot)$ and extract $\boldsymbol{h}_{\text{last}}$, $\boldsymbol{h}_{\text{verb}}$, and $\hat{y}$. Each notation refers to a mean-pooled last layer hidden representation $\boldsymbol{h}_{\text{last}}$, the verbalizer probability distribution of the `[Mask]` token $\boldsymbol{h}_{\text{verb}}$, and its corresponding pseudo-label $\hat{y}$. CeLDA employs two different representations the $\boldsymbol{h}_{\text{last}}$ and $\boldsymbol{h}_{\text{verb}}$ since they retain complementary information which might be beneficial in learning a precise decision boundary. Namely, the former encapsulates rich information, including semantics and syntactic information, and the latter possesses specific probability distribution of `[Mask]` token. To maximize the capability of $\boldsymbol{h}_{\text{verb}}$, we expand the verbalizer in Eq. 1 into predefined multiple words following Hu et al. (2021). Finally, we concatenate both representations to derive a single final representation $\boldsymbol{h}$:

$$\boldsymbol{h} = [||\boldsymbol{h}_{\text{last}}||_2 \oplus \boldsymbol{h}_{\text{verbal}}]. \qquad (3)$$

We apply L-2 normalization to $\boldsymbol{h}_{\text{last}}$ before concatenation to synchronize the range with the representation provided by the verbalizer (ranging from 0 to 1). Finally, we construct the pseudo labeled dataset $\mathcal{D}^{\text{pseudo}} = \{\boldsymbol{h}_i, \hat{y}_i\}_{i=1}^k$, for the next training phase. After this pre-processing step, we no longer use the LM, which tremendously reduces computational cost in training.

4.2 Data Cleansing with Cluster Entropy

The objective of this training phase is to filter out the uncertain data samples in $\mathcal{D}^{\text{pseudo}}$ to create a more reliable dataset, $\mathcal{D}^{\text{clean}}$. To achieve this goal, we utilize clustering on the representations $\boldsymbol{h}$ based on the previous findings that the LM's features are grouped with semantically similar sentences. Specifically, we employ KMeans to cluster the pseudo-labeled dataset $\mathcal{D}^{\text{pseudo}}$ into K clusters. And let $\boldsymbol{h}_{ij}$ be the representation belonging to the i^{th} cluster K_i and j be the index of the sample in the cluster. From each cluster, we estimate the pseudo-label probability distribution within each cluster $P_{k_i}(\mathcal{Y})$:

$$P_{K_i}(\mathcal{Y} = y) = \frac{|\mathcal{S}|}{|K_i|},$$

where $\mathcal{S} = \{s \in \hat{y}_i | \hat{y}_i = y\}$ and i, y denotes cluster index and label index, respectively.

Then we measure the entropy weight (EW) of each cluster to estimate their uncertainty:

$$\text{EW}(K_i) = \frac{1 - \text{NormEnt}(K_i)}{\sum_{j \in K}(1 - \text{NormEnt}(K_j))}, \qquad (4)$$

$$\text{NormEnt}(K_i) = -\frac{\sum_{l \in \mathcal{Y}} p_{K_i}(l) \log p_{K_i}(l)}{\log |C|}.$$

EW increases when the samples within a cluster tend to have the same label, but decreases otherwise. By setting a threshold, τ, we are able to select a portion of certain clusters K^{clean} while removing several uncertain clusters.

$$K^{\text{clean}} = \{K_i \in K | \, \text{EW}(K_i) \geq \tau\} \qquad (5)$$

Furthermore, we eliminate samples in the cluster that do not conform to the majority pseudo labels within the cluster and create the final certain dataset $\mathcal{D}^{\text{clean}}$:

$$\mathcal{D}^{\text{clean}} = \{K_i \in K^{\text{clean}} | \hat{y}_{ij} = \arg\max_{y \in \mathcal{Y}} P_{K_i}(\mathcal{Y})\} \qquad (6)$$

And we utilize the $\mathcal{D}^{\text{clean}}$ to train LDA. Figure 2 illustrates § 4.1, 4.2 stage of CELDA.

4.3 LDA Training

Utilizing the filtered dataset $\mathcal{D}^{\text{clean}}$ from prior step, we finally fit the parameters of LDA through MLE as introduced in § 3.3. To further improve LDA, we recursively train LDA by updating the pseudo labels with the trained model and repeating the previous data cleansing steps based on the assumption that the trained LDA produces more precise pseudo labels. To prevent the model from oscillating or diverging in iterative training, we employ moving average (MA):

$$\hat{\boldsymbol{\mu}}_{t+1} = \frac{t-1}{t}\hat{\boldsymbol{\mu}}_{t-1} + \frac{1}{t}\hat{\boldsymbol{\mu}}_t,$$

$$\hat{\Sigma}_{t+1} = \frac{t-1}{t}\hat{\Sigma}_{t-1} + \frac{1}{t}\hat{\Sigma}_t,$$

where t indicates timestamp (current epoch). If the updated label does not deviate beyond a specific ratio δ from the previous label, we terminate the training process judging the model has converged. The overall procedure of CELDA is summarized in Algorithm 1.

Algorithm 1: CELDA Training.

```
Input: unlabeled dataset $\mathcal{D}$, a language
       model LM, pre-defined verbalizer $\mathcal{V}$,
       LDA model $M$
Step 1: Pre-processing
$\mathcal{D}^{\text{p}} = \{\}$
for $\boldsymbol{x} \in \mathcal{D}$ do
    1a. get embedding $\boldsymbol{h} \leftarrow$ Eq.3;
    2a. get pseudo-label $\hat{y} \leftarrow$ Eq.2;
    $\mathcal{D}^{\text{p}} = \mathcal{D}^{\text{p}} \cup (\boldsymbol{h}, \hat{y})$;
Return $\mathcal{D}^{p}$;
Step 2: Data Cleansing & LDA training
for until convergence. do
    1b. run KMeans on $\mathcal{D}^{p}$ with $k = K$
        until converge;
    2b. data cleansing $\mathcal{D}^{c} \leftarrow$ Eq.5, 6;
    3b. train $M$ with $\mathcal{D}^{c}$;
    for $(\boldsymbol{h}, \hat{y}) \in \mathcal{D}^{p}$ do
        4b. $\hat{y} = M(\boldsymbol{h})$ ←update label ;
Output: A trained LDA classifier $M$.
```

As previously discussed in § 3.3, LDA is a variation of Gaussian Discriminative Analysis (GDA) that utilizes shared covariance among classes, resulting in a reduction of parameters from $O(|\mathcal{Y}|d^2)$ to $O(|\mathcal{Y}|d)$. Despite the potential negative impact on performance compared to GDA when data is abundant and clean, LDA possesses several beneficial properties, such as the ability to fit the model with fewer samples and greater robustness to noisy labels (Murphy, 2022). Thus, the reduced parameters in LDA prevent overfitting and improve the model's ability to adapt to the test dataset, in contrast to other Gaussian-based approaches that are prone to overfitting and yields a poor performance in test cases, as will be demonstrated in our following experiments. The mentioned characteristics of

LDA are highly valuable in our scenario, where the training dataset is noisy and a portion of the available data is discarded during the previous data cleansing stage.

5 Experiments

5.1 Backbone & Datasets

We adopt T5 (Raffel et al., 2020) as the main backbone of our experiments which is fairly large (upto 11 billion parameters) and open-sourced. Furthermore, we report additional experimental results with SimCSE (Gao et al., 2021) supervised RoBERTa-large (Liu et al., 2019) in the Appendix. To investigate the performance of each method in many different scenarios, we carefully select 8 datasets [1]: AGNews, DBPedia, IMDb, SST2, Amazon, Yahoo, Banking77, and CLINC. The statistics of datasets used in the experiments are reported in Table 1. To evaluate each method in stable conditions, we report the average accuracy of 5 different seeds (13, 27, 250, 583, 915) along with the corresponding standard deviation of 5 runs as a model performance.

5.2 Experimental Configurations

For all experiments, we utilize KMeans clustering with *euclidean* distance, and set the number of clusters to $K = |\mathcal{Y}| \times 16$ except binary classification task. For binary classification tasks, we encountered an issue where the number of clusters became significantly smaller compared to the size of the total dataset. As a solution, we opted to use a relatively large value for $K = |\mathcal{Y}| \times 64$. Furthermore, we set maximum tokenizer length and exit threshold δ adaptively depending on the dataset size. And we utilized the same template and verbalizer for every task from Openprompt (Ding et al., 2021). Detailed configurations (i.e., templates, verbalizer) and our computation environments are stipulated in the Appendix.

5.3 Competing Methods

We compare our methods with state-of-the-art zero-shot text classification methods and several baseline fully supervised methods:

- **LDA (full)**: We report the accuracy of the baseline LDA model trained on the top of LM representations with a fully-annotated dataset, which serve as our upper bound.

Datasets	# Train	# Test	# Cls
DBPedia	560,000	70,000	14
Yahoo	1,400,000	60,000	10
AGNews	120,000	7,600	4
SST2	6,920	1,821	2
Amazon	3,600,000	400,000	2
IMDb	25,000	25,000	2
Banking77	10,003	3,080	77
CLINC	15,250	550	150

Table 1: Dataset statistics

- **PET**: Pattern-Exploiting Training (Schick and Schütze, 2021) is a baseline zero-shot prompting method that transforms an input into a cloze-task and utilizes a single-word verbalizer to label data sample.
- **KPT**: Knowledge Prompt-Tuning (KPT) (Hu et al., 2022) is a multi-word verbalizer expansion of PET that extracts multi-words similar to the original label name from external knowledge bases.
- **SimPTC**: (Fei et al., 2022b) iteratively trains a Gaussian Mixture Model (GMM) on top of the LM's representations from the initial pseudo labels with E&M algorithm, which measures the similarity between the anchor embedding and data sample.

5.4 Main Results

Table 2 reports the performance of CeLDA and other competing methods on 6 benchmarks with 3 LMs of varying size (770M to 11B). From the results, we share following observations:

(1) Significant performance of CeLDA: Our method consistently outperforms other competing methods by a large margin in varying language models. Additionally, another black-box WSL method (SimPTC) often fails to reach a stable convergence point and performance drops as the training continues[2]. This is because they estimate the class-specialized covariance matrix resulting in exponential growth of the model parameter. Thus SimPTC's decision boundary does not align well with the test distribution meaning that the model is prone to overfitting. We delve into this phenomenon in detail in the Appendix. However, our method constantly exhibits better performance and

[1] The detailed descriptions and references of each dataset are stipulated in the Appendix.

[2] Experimental setup in SimPTC paper utilizes both train and test dataset in training stage, unlike ours.

			T5-Large (770M)				
Setting	**Method**	**DBPedia (14)**	**Yahoo (10)**	**AGNews (4)**	**SST2 (2)**	**Amazon (2)**	**IMDb (2)**
Full	LDA	98.72	72.40	91.37	92.09	96.05	94.28
ZSL	PET	62.14 ± 0.0	34.00 ± 0.0	54.37 ± 0.0	69.96 ± 0.0	78.47 ± 0.1	77.96 ± 0.0
	KPT	81.69 ± 0.6	62.32 ± 0.4	84.79 ± 0.8	82.70 ± 0.4	87.22 ± 0.7	83.71 ± 0.6
WSL	SimPTC	68.49 ± 0.1	47.9 ± 0.1	87.01 ± 0.0	68.48 ± 0.0	**94.75** ± 0.1	**92.74** ± 0.0
	CELDA	**84.47** ± 0.3	**68.88** ± 0.1	**90.03** ± 0.2	**89.40** ± 0.5	94.70 ± 0.1	91.70 ± 0.1
			T5 (3B)				
Full	LDA	98.75	73.73	92.11	94.23	96.74	95.35
ZSL	PET	62.11 ± 0.0	32.07 ± 0.0	46.36 ± 0.0	70.29 ± 0.0	68.84 ± 0.1	77.38 ± 0.0
	KPT	82.78 ± 0.2	62.17 ± 0.2	86.03 ± 0.3	86.05 ± 0.1	84.15 ± 0.5	81.93 ± 1.3
WSL	SimPTC	68.60 ± 0.2	50.20 ± 0.3	87.54 ± 0.1	89.79± 0.1	95.99 ± 0.1	94.46 ± 0.0
	CELDA	**85.11** ± 1.5	**70.35** ± 0.1	**90.18** ± 0.3	**92.42** ± 0.1	**96.08** ± 0.1	**94.55** ± 0.0
			T5 (11B)				
Full	LDA	98.87	73.64	92.21	94.51	97.03	95.88
ZSL	PET	65.47 ± 0.0	39.66 ± 0.0	62.51 ± 0.0	71.00 ± 0.0	71.58 ± 0.1	71.78 ± 0.0
	KPT	83.45 ± 1.0	63.34 ± 0.2	86.09 ± 0.6	86.46 ± 0.1	88.03± 0.4	84.08 ± 1.2
WSL	SimPTC	69.83 ± 0.1	51.02 ± 0.5	88.61 ± 0.1	88.58 ± 0.0	96.42 ± 0.0	95.02 ± 0.0
	CELDA	**86.88** ± 0.3	**71.38** ± 0.6	**90.29** ± 0.3	**94.23** ± 0.4	**96.78** ± 0.0	**95.61** ± 0.0

Table 2: Experimental results on 6 benchmarks. Best method (except full LDA) for each dataset is indicated in **bold**. Full, ZSL, and WSL indicates fully supervised, zero-shot learning, and weakly-supervised learning, respectively. Our methods surpasses other compared methods and close the gap with fully supervised method.

can capture precise decision boundaries without overfitting. An additional noteworthy observation is that CELDA approaches the fully supervised LDA with a small spread on most benchmarks.

(2) Scalability of CELDAto large models: Additionally, our method demonstrates scalability to large models, meaning the performance of CELDA improves as the scale of the LM grows. Other methods are not scalable since the performance of ZSL method easily saturates and displays insignificant improvement, and other well-performing WSL methods often require explicit model training. However, the performance of CELDA method improves with the larger language model, which indicates that our method exploits the high quality representations from the larger model quite properly.

6 Analysis

We conduct an in-depth investigation of CELDA to elucidate its underlying mechanism in conjunction with our intuitions. For all analysis experiments, we utilize T5-large as a backbone model.

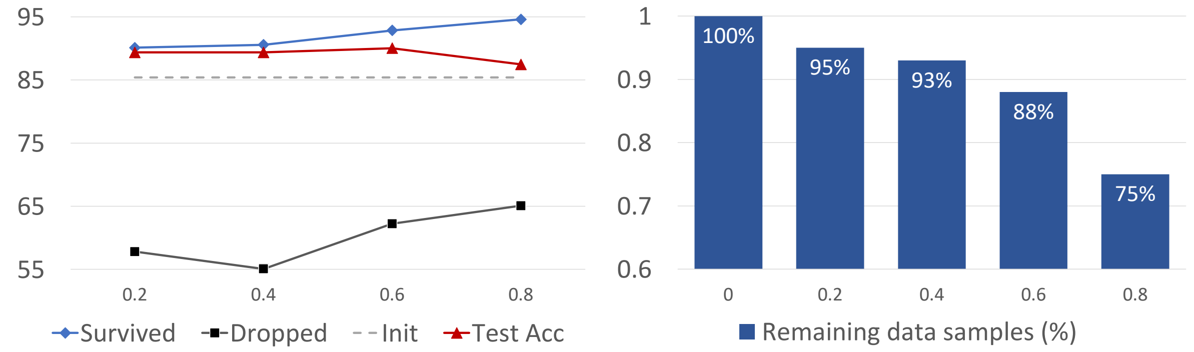
Model	Accuracy
Vanilla Prompting (PET)	80.96
+ LDA	84.49
+ Recursive train	85.45
+ multi-verbalizer	88.10
+ feature augmentation	88.57
+ data cleansing (CELDA)	**90.03**

Table 3: Component-wise ablation study in CELDA

6.1 Ablations

We carry out ablation studies in several aspects to further explore the effectiveness of the main components in our approach:

Component ablation: To validate our model design, we conducted a component-wise ablation study on the AGNews dataset, where we sequentially added each component in CELDA. From Table 3, we confirmed that our components improve the overall performance progressively. Specifically, as mentioned § 3.3, LDA learns a more precise decision from a noisy labeled dataset, and recursive



(a) Accuracy of filtered & abandoned datasets.

(b) Remaining portion of dataset after filtering.

Figure 3: Filtered dataset has more accurate data samples while the accuracy of dropped samples exhibits poor performance. Setting a high strong threshold increases the pseudo label accuracy but the overall performance drops.

training also gives marginal improvement. Moreover, utilizing a multi-word verbalizer as in Hu et al. (2022) and augmenting their logit representation increases the overall performance. Finally, applying our clustering-based data cleansing approach improves the performance further and yields the best performance.

Effectiveness of data cleansing: We scrutinize our clustering-based filtering module in various aspects to verify its efficacy. Firstly, we compare the quality of both datasets: One that has survived after the data filtering stage (survived dataset for abbreviation) and the other that has been dropped (dropped dataset for short). Figure 3a illustrates the accuracy when the cluster filtering threshold τ changes. From the figure, we verified that our survived dataset has more accurate data samples, while the accuracy of dropped samples lags far behind survived dataset. Moreover, setting a stronger threshold τ increases the pseudo-label accuracy of the survived dataset and reaches a near-clean dataset, but the overall performance drops. It implies that giving strong conditions discards even meaningful samples, which is beneficial in training. As a support, we can verify that the accuracy of the dropped dataset also increases with a higher τ, and the total number of the survived sample decreases, as shown in Figure 3b.

6.2 Impact of Initial Pseudo-labels

Similar to other WSL methods, our methodology is heavily influenced by initial pseudo-labels. LMs generate highly reliable pseudo labels for coarsely labeled datasets, as shown in Figure 2. However, labeling a fine-grained dataset with LM often leads to poor performance, even with recent ZSL methods such as (Wang et al., 2021; Meng et al., 2020), only taking laborious manual engineering. Meanwhile, the hidden representations from LM have the potential to discriminate this fine-grained dataset without adaptation. For instance, in Banking dataset (77 classes) and CLINC dataset (150 classes), existing zero-shot labeling methods and WSL methods, including CELDA, output unsatisfactory performances as seen in 4.

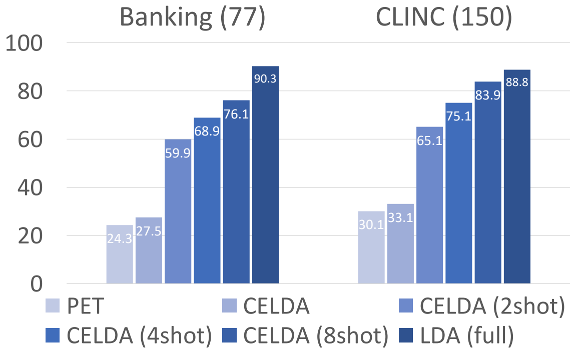


Figure 4: Experiments on fine-grained datasets. With a few true labels, our methods improves drastically.

CELDA can address this limitation by incorporate concepts from active learning (AL) which annotates a few selected samples with true labels through a human-in-the-loop pipeline. Specifically, we annotate n-shot samples per class (total $n \times |\mathcal{Y}|$ samples) that are closest to each centroid. Then, we annotate whole sentences in each cluster with the true label from the closest sample and re-train LDA as usual. While this selection approach is quite simple, it selects highly-meaningful samples from the unlabeled dataset, significantly improving the performance of CELDA sharply with min-

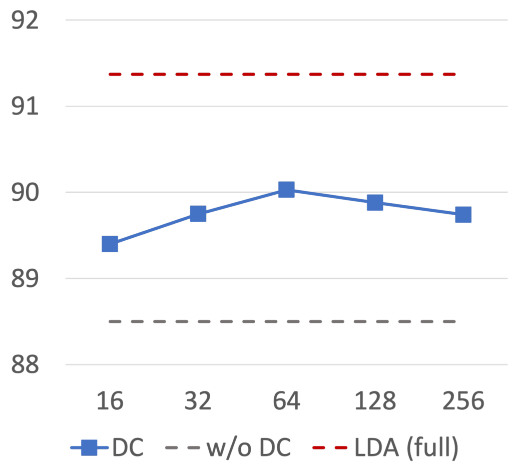


Figure 5: Correlation between the number of clusters and the performance.

imal human annotated samples. As depicted in Figure 4, we revealed that labeling 8 shots per class on fine-grained datasets significantly enhances the performance (by nearly 50% on average), where traditional methods tend to struggle.

6.3 Number of Clusters

We conducted an additional investigation of CELDA to analyze its the underlying mechanism. We utilize T5-large as a backbone mode and tested on AGNews dataset. Figure 5 illustrates the correlation between the number of clusters and the performance. We can confirm that the performance improves as the number of clusters increases, but slowly deteriorates when the number of clusters increases too much. As the total number of cluster increases, the samples belonging to each cluster decreases. Accordingly, entropy weight estimated from Eq. 4 becomes unreliable hurting the effectiveness of the overall data cleansing process. Based on this result, we set number of clusters to $16 \times |\mathcal{Y}|$.

7 Conclusion

This work presents CELDA, a practical framework for employing a black-box language model. We have sought room for improvement in three orthogonal directions: (1) Utilizing language models with high-quality representations (from the last layer and logit distribution). (2) Filtering unreliable data samples from the noisy dataset. (3) Recursively trains LDA, which is robust to noisy samples and avoids over-fitting by minimizing the overall model parameters. By fusing these elements, we demonstrate the significant performance of CELDA on sundry classification tasks and its scalability with the language model size. In our follow-up study, we aim to employ sample-wise entropy from pseudo-labeling in the data cleansing phase instead of utilizing the entropy of the cluster, which is highly course-grained. We expect that looking at fine-grained sample-wise entropy can yield a more precise data filtering effect, reducing meaningful samples from being dropped.

8 Limitations

While our method demonstrates strong performance in our experimental setups, potential issues may arise when the characteristics of the available unlabeled dataset drastically change. For one example, if the scale of the available dataset is too small, the effectiveness of our clustering-based data filtering may fall drastically, leading to poor performance. Or, if the dataset is highly unbalanced, our model cannot acquire information about several specific classes. One way to compensate for this shortcoming is to use an externally imported corpus or dataset, similar to other ZSL or WSL methods. Another drawback of CELDA is that the final performance is highly dependent on the performance of the initial pseudo label, as shown in ablation. Nevertheless, as demonstrated in our ablation studies, we can remedy this issue by labeling a few samples, like active learning.

9 Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)]

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

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.

Martin Brümmer, Milan Dojchinovski, and Sebastian Hellmann. 2016. Dbpedia abstracts: a large-scale, open, multilingual nlp training corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3339–3343.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Hyunsoo Cho, Hyuhng Joon Kim, Junyeob Kim, Sang-woo Lee, Sang-goo Lee, Kang Min Yoo, and Taeuk Kim. 2023. Prompt-augmented linear probing: Scaling beyond the limit of few-shot in-context learners. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI*.

Shizhe Diao, Xuechun Li, Yong Lin, Zhichao Huang, and Tong Zhang. 2022. Black-box prompt learning for pre-trained language models. *arXiv preprint arXiv:2201.08531*.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

Yu Fei, Ping Nie, Zhao Meng, Roger Wattenhofer, and Mrinmaya Sachan. 2022a. Beyond prompting: Making pre-trained language models better zero-shot learners by clustering representations. *arXiv preprint arXiv:2210.16637*.

Yuxiao Fei, Ping Nie, Zhao Meng, Roger Wattenhofer, and Mrinmaya Sachan. 2022b. Beyond prompting: Making pre-trained language models better zero-shot learners by clustering representations. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Jimin Hong, Jungsoo Park, Daeyoung Kim, Seongjae Choi, Bokyung Son, and Jaewook Kang. 2022. Tess: Zero-shot classification via textual similarity comparison with prompting using sentence encoder. *arXiv preprint arXiv:2212.10391*.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. 2021. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035*.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing EMNLP*.

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*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL*.

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. Z-icl: Zero-shot in-context learning with pseudo-demonstrations. *arXiv preprint arXiv:2212.09865*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. 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 (EMNLP)*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Kevin P Murphy. 2022. *Probabilistic machine learning: an introduction*. MIT press.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in Neural Information Processing Systems*, 34:11054–11070.

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

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, EACL*.

Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. 2022. Nearest neighbor zero-shot inference. *arXiv preprint arXiv:2205.13792*.

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

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning, ICML*.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021. 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*.

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

Lu Zhang, Jiandong Ding, Yi Xu, Yingyao Liu, and Shuigeng Zhou. 2021. Weakly-supervised text classification based on keyword graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML*.

Appendix

A Dataset Description

In our experiments we use 8 different benchmark datasets which include topic classification, intent classification, question classification, and binary sentiment classification datasets.

DBpeida (Brümmer et al., 2016) is an ontology classification dataset with DBpedia documents and 14 topics. It is a balanced dataset containing 40,000 training data and 5,000 testing data per class.

Yahoo (Zhang et al., 2015) dataset is composed of a pair of questions and answers and a topic of it.

AGNews (Zhang et al., 2015) is a news topic classification dataset from AG's news corpus with 4 different classes.

IMDb (Maas et al., 2011) is a binary movie review dataset for sentiment classification.

SST-2 (Socher et al., 2013) is for detecting the sentiment of a single sentence of the movie review.

Amazon (McAuley and Leskovec, 2013) is a review dataset Amazon from various domains (e.g., electronic stuff) for sentiment classification.

Banking77 (Casanueva et al., 2020) a intent classification dataset which comprises fine-grained 77 intents in a single *banking* domain regarding customer service queries.

CLINC (Larson et al., 2019) is for classifying an intention of queries in dialog systems. The classes of CLINC dataset cover a total of 150 classes in 10 different domains and one out-of-scope class.

B Detailed Implementation Details

In pseudo label initialization of training data, a zero-shot prediction ability of T5 is utilized with a template and verbalizers. Expanded label words are used as verbalizers in experiments of main datasets for rich logit representations and pseudo labels. We also take a template with a mask to get mask logit values of pre-defined verbalizers from the pre-trained T5 model.

Templates and Verbalizers: We apply manual templates from OpenPrompt (Ding et al., 2021) in zero-shot pseudo-labeling which are listed in Table 4. To annotate samples with pseudo-labeling, expanded label words constructed by KPT (Hu et al., 2021) are employed for each task (see Table 5). For Banking77 and CLINC dataset, we use true label words without any expansion due to their abundant classes (77 and classes).

Environments and Utilized Libraries: We utilize 8 RTX A6000 (48GB) GPUs for the experiments. When we extract initial pseudo labels and representations, KPT framework (Hu et al., 2021) in a zero-shot setting is used with OpenPrompt (Ding et al., 2021) library. Among various sizes of T5 models, we utilize T5-large, T5-3B, and T5-11B from Transformers (Wolf et al., 2020) library in Huggingface. The batch size is set adaptively depending on the average length of each dataset and the size of T5. KMeans code from https://github.com/subhadarship/kmeans_pytorch library is also used in CELDA.

C Comparison with SimPTC

As an implementation detail of SimPTC (Fei et al., 2022a), both the train and test datasets are used in fitting Bayesian Gaussian Mixture Model (BGMM) by considering them as a set of unlabeled data. Then, SimPTC measures the accuracy of the test dataset, which is a portion of the unlabeled dataset used in training GMM. It is different from our setting of using only a train dataset to train LDA. Thus we additionally experiment with the setting of SimPTC.

Since SimPTC mainly uses SimCSE (Gao et al., 2021) supervised RoBERTa large (Liu et al., 2019) embeddings in their experiments, we also extract embeddings and construct pseudo labels of samples with SimCSE supervised RoBERTa large. SimCSE supervised RoBERTa significantly loses its ability of predicting masked words while fine-tuned with contrastive objective without MLM. Thus we could not perform mask prediction based zero-shot pseudo labeling with SimCSE supervised RoBERTa large. Instead, we initialize pseudo labels with Encode & Match, a process of generating pseudo-label, proposed by SimPTC, which assign a pseudo-label to each input embedding with class anchor sentence embeddings. Consequently, initial pseudo labels of samples are identical for both CELDA and SimPTC.

We follow SimPTC's design of the experiment in reproducing its performance. The experiment with CELDA is performed without the representation augmentation since verbalizer logit representation is unavailable.

According to the results in Table 6, CELDA outperforms SimPTC in most of the experiments. Especially, CELDA displays a better accuracy in most cases when we utilize only a train split dataset

Task type	**Dataset**	**Template**
Sentiment	IMDb, SST2, Amazon	It was [MASK]. [input sentence]
Topic	DBPedia Yahoo AGNews Banking77, CLINC	[input sentence] is a [MASK]. A [MASK] question : [input sentence] A [MASK] news : [input sentence] [Category : [MASK]] [input sentence]

Table 4: Pre-defined templates for each task.

Dataset	**Verbalizer (True label word: Expanded words)**
IMDb, SST2, Amazon	negative: bad,abysmal,adverse,alarming,angry,annoy, anxious, worthless,wound,yell,yucky, ... positive: good,absolutely,accepted,acclaimed,accomplish, wealthy,welcome,well,whole, ...
DBPedia	company: corporation, company, corp, shareholder, enterprise, conglomerate, firm, ... school: school, education, university, academy, college, teacher, classroom, ... book: novel, publication, book, fiction, publishing, author, prose, magazine, text, novella, ...
Yahoo	society: society, culture, civilization, philosophy, association, anthropology, guild, subculture, ... science: science, mathematics, biology, mathematician, scientist, calculus, geometry, ... politics: politics, government, governance, administration, law, democracy, aristotle, state, ...
AGNews	politics: politics,government,diplomatic,law,aristotle,diplomatical,governance,republic, ... sports: sports,athletics,gymnastics,sportsman,competition,cycling,soccer,tennis,game, ... technology: technology,engineering,science,biotechnology,internet,nanotechnology, ...

Table 5: Pre-defined verbalizers for each task.

which is a usual case in machine learning. Even in using both train and test datasets as an unlabeled dataset, CELDA outperforms SimPTC in most of the results. While LDA in CELDA shares covariance among classes, GMM in SimPTC computes a covariance matrix for each class. It causes overfitting on training samples and results in poor performance in test cases. For CELDA, the overall performance of two different settings are stable with a tied-covariance. Since SimPTC uses tied-covariance setting for IMDb and Amazon datasets, the performance of SimPTC and CELDA are close in both cases.

Split	**Method**	**DBPedia (14)**	**Yahoo (10)**	**AGNews (4)**	**IMDb (2)**	**SST2 (2)**	**Amazon (2)**
Train	SimPTC	67.22 ± 0.5	50.60 ± 0.1	85.53 ± 0.4	**88.13** ± 0.0	89.99 ± 0.3	94.46 ± 0.2
	CELDA	**83.43** ± 0.1	**61.34** ± 0.9	**88.82** ± 0.5	87.11 ± 0.1	**90.18** ± 0.2	**94.50** ± 0.1
+ Test	SimPTC	80.43 ± 0.0	**63.13** ± 0.0	85.82 ± 0.1	**88.47** ± 0.0	88.8 ± 0.2	94.38 ± 0.1
	CELDA	**83.47** ± 0.2	62.62 ± 0.1	**88.09** ± 0.4	87.62± 0.1	**89.91** ± 0.1	**94.39** ± 0.1

Table 6: Experimental results on 6 datasets with SimCSE-Roberta-Large. While our method constantly exhibits similar performance with or without a test split, SimPTC's decision boundary does not align well without the test split, meaning that the model is prone to overfitting.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section 8

☑ A2. Did you discuss any potential risks of your work?
section 8

☑ A3. Do the abstract and introduction summarize the paper's main claims?
abstraction, 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 5,6

☑ B1. Did you cite the creators of artifacts you used?
appendix A

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
appendix A

□ 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?
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.
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?
section 5 appendix c

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 5 appendix c

☑ 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 5

☑ 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 c

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.?
No response.

□ 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)?
No response.

□ 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?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.