Modeling Data Diversity for Joint Instance and Verbalizer Selection in Cold-Start Scenarios

Mohna Chakraborty(✉), Adithya Kulkarni, and Qi Li

Iowa State University, Ames, IA 50011, USA
{mohnac,aditkulk,qli}@iastate.edu

Abstract. Prompt-based methods leverage the knowledge of pre-trained language models (PLMs) trained with a masked language modeling (MLM) objective; however, these methods are sensitive to template, verbalizer, and few-shot instance selection, particularly in cold-start settings with no labeled data. Existing studies overlook the dependency between instances and verbalizers, where instance-label probabilities depend on verbalizer token proximity in the embedding space. To address this, we propose ColdSelect, a joint verbalizer and instance selection approach that models data diversity. ColdSelect maps PLM vocabulary and $h_{[MASK]}$ embeddings into a shared space, applying dimensionality reduction and clustering to ensure efficient and diverse selection. By optimizing for minimal uncertainty and maximal diversity, ColdSelect captures data relationships effectively. Experiments on eight benchmarks demonstrate ColdSelect superiority in reducing uncertainty and enhancing generalization, outperforming baselines in verbalizer and few-shot instance selection for cold-start scenarios.

Keywords: Cold-start setting · Prompt-based Learning · Data Diversity Modeling

1 Introduction

Pre-trained language models (PLMs) trained with the masked language modeling (MLM) objective [18] have become essential for various NLP downstream tasks [2], as their training on extensive corpora allows them to capture rich contextual information. Prompt-based methods capitalize on this by transforming classification tasks into cloze-style tasks [6], where PLMs predict the [MASK] token using suitable vocabulary tokens. This alignment with the pre-training objective allows prompt-based methods to deliver strong performance, even with limited labeled data. In this study, we focus on moderately sized masked language models, as generative models pose challenges such as high deployment

M. Chakraborty and A. Kulkarni—The first two authors contributed equally to this work.

X. Wu et al. (Eds.): PAKDD 2025, LNAI 15870, pp. 461–477, 2025.
https://doi.org/10.1007/978-981-96-8170-9_36

costs on local hardware and privacy concerns when using APIs [1] for sensitive data. Our approach balances efficiency, performance, and data security.

Prompt-based methods rely on two key components, templates and verbalizers, that together unlock the potential of PLMs for downstream tasks. Templates reframe input data into cloze-style tasks, enabling the model to leverage its pretrained MLM capabilities, while verbalizers map the model's vocabulary predictions to class labels, serving as the crucial link between token outputs and task-specific categories. Templates can be manually designed [28,35], automatically generated [6,17], or constructed continuously [12,14,15]. Similarly, verbalizers can be divided into three categories: manual [28,29], search-based [6,27,32], and soft verbalizers [8,41]. While the manual creation of templates and verbalizers provides a straightforward approach, it is inherently limited by human interpretation, often resulting in suboptimal representations. In contrast, automatic and continuous methods reduce manual effort, dynamically adapting to optimize the model's performance. However, the effectiveness of prompt-based methods remains highly sensitive to the choice of templates [3], verbalizers [6], and few-shot labeled instances [39]. This sensitivity underscores the need for approaches that better model the diversity and complexity of data distributions to ensure robust and generalizable performance.

To enhance the performance of prompt-based methods, we focus on annotating instances and obtaining verbalizer tokens within a given labeling budget [11]. Efficient use of this budget requires a balanced approach to both instance and verbalizer selection, as these elements are interrelated. Ignoring this relationship can result in suboptimal outcomes. Existing methods for verbalizer selection [6,36] rely on randomly chosen few-shot instances, often lacking the diversity needed for robust generalization. Similarly, instance selection approaches [39] using fixed, manually designed verbalizers fail to capture data variability or adapt to nuanced label distributions. These studies overlook the dependency between instance and verbalizer selection. Under the MLM objective, an instance is more likely to predict a label accurately if the verbalizer token lies nearby in the embedding space. Ignoring this relationship results in redundant examples, noisy data, and outliers, which degrade generalization and robustness, especially in cold-start scenarios without labeled data.

To address the aforementioned challenges of data diversity and uncertainty in cold-start scenarios, we propose ColdSelect, a novel method that jointly selects verbalizers and few-shot instances by modeling data diversity. Modeling data diversity ensures the selection of diverse instances that represent the corpus comprehensively, reducing redundancy and improving generalization. For example, in sentiment analysis tasks, including instances with varying sentiment intensities helps the model learn nuanced distinctions, while in news classification, diverse examples across categories ensure balanced representation. At the same time, diverse verbalizer tokens for a class, such as mapping "great" and "magnificent" to *"positive"* class, effectively capture label semantics and avoid oversimplified mappings. Jointly optimizing instance and verbalizer selection within a single labeling budget maximizes efficiency and minimizes noise and redundancy.

ColdSelect maps pre-softmax embeddings of PLM vocabulary tokens and $h_{[MASK]}$ embeddings into a shared space for efficient comparison. Dimensionality reduction using PCA [38] enhances computational efficiency, while clustering methods like KMeans [20] and refinement with negative silhouette loss [26] ensure robust, well-separated clusters that capture the data's diversity. Within the resulting clusters, instance and verbalizer selection are guided by an optimization framework that operates under a labeling budget $\mathcal{B}$. This framework minimizes labeling uncertainty at each step by balancing three critical factors: intra-cluster cohesion, which ensures selected instances are representative; inter-cluster separation, which avoids redundancy across clusters; and impurity, which captures label diversity. At each timestamp, ColdSelect identifies the most informative clusters, from which the instances to annotate and verbalizer tokens are selected. By integrating these steps, ColdSelect ensures that the selected tokens and instances reflect the dataset's diversity and maximize the model's generalization capability, ultimately improving performance in prompt-based tasks.

In summary, the contributions of this study are as follows:

1. To the best of our knowledge, this is the first method to jointly and automatically select instances to annotate and verbalizer tokens in a cold-start setting. By modeling data diversity using shared embedding spaces, clustering techniques, and a novel selection-based optimization framework, our approach ensures robustness and generalization, effectively addressing sensitivity in prompt-based methods.
2. The instance and verbalizer selection process is formulated as an optimization problem designed to minimize labeling uncertainty at each step of the selection process.
3. Comprehensive experiments on benchmark datasets show that ColdSelect successfully models data diversity and the selected instances and verbalizer tokens reduce labeling uncertainty, leading to improved accuracy.

2 Related Works

Despite the remarkable success of PLMs, their application in specific tasks remains challenging, particularly in cold-start scenarios where no labeled data is available. This limitation has led to the growing interest in prompt-based methods, which reformulate downstream tasks into cloze-style tasks to better align with the MLM pre-training objective. Prompt-based approaches rely on three key components: templates, verbalizers, and, optionally, a few labeled instances for fine-tuning. While significant progress has been made in automatic and continuous template generation [6,12,14,15,17,28,35], the automatic selection of verbalizers and few labeled instances remains underexplored, particularly in cold-start settings. Below, we review related work in these two areas.

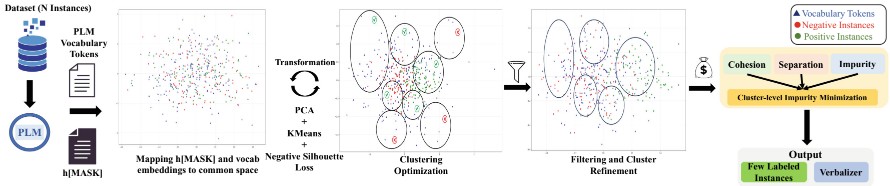

Fig. 1. Overview of ColdSelect

Few-Shot Instance Selection in Cold-Start Settings. Selecting diverse and representative few-shot instances is essential for improving the performance of prompt-based methods, particularly in cold-start scenarios where only unlabeled data is available. Early approaches [23] relied on clustering and heuristic-driven selection, but their inability to account for inter-sample diversity limited their effectiveness. Subsequent methods [4,40] utilized PLMs, leveraging embedding spaces or MLM loss to guide instance selection. While these strategies were task-agnostic, they often struggled with misalignment between pre-training objectives and downstream tasks, leading to suboptimal results. Recent efforts [16,34] have focused on few-shot selection for large-scale language models through in-context learning. However, these methods lack a cohesive framework to simultaneously address data diversity and labeling uncertainty, leaving significant room for improvement.

Verbalizer Selection. Verbalizers play a crucial role in mapping model predictions to class labels. Early approaches [28,29] relied on manually designed verbalizers, which, while effective, were time-consuming and susceptible to human bias. To automate this process, search-based methods [27,32] identified tokens that maximized conditional probabilities within the LLM vocabulary. However, these approaches often generated tokens that lacked contextual relevance. Enhancements using semantically similar tokens from external knowledge bases [10] improved token quality but failed to address data diversity and struggled with scalability in large vocabularies and few-shot scenarios. Soft verbalizers [8,41] mitigated some of these limitations by learning continuous embeddings but required substantial labeled data, making them unsuitable for few-shot settings. More recently, prototypical verbalizers [5] leveraged few-shot training data to generate prototype embeddings, achieving state-of-the-art performance in automated verbalizer design. However, even these methods often fell short of manual verbalizers in certain cases, highlighting the need for further improvement. More recently, [36] proposed the tuning-free LLE-INC method, re-embedding the verbalizer space using intra-class neighborhood relationships to enhance the design.

Unlike previous studies, ColdSelect is the first approach to jointly select instances and verbalizers while explicitly modeling their interdependence and data diversity in cold-start settings.

3 Preliminaries

In this section, we describe the process of obtaining prediction probabilities in prompt-based learning. Given a template $\mathcal{T}$, a verbalizer $\mathcal{M} : \mathcal{Y} \rightarrow \mathcal{V}$ that maps the class label space $\mathcal{Y}$ to tokens in the PLM vocabulary $\mathcal{V}$, and an input instance $\mathcal{I}$ from the unlabeled corpus $\mathcal{D}$, the probability of $\mathcal{I}$ being assigned a label $y \in \mathcal{Y}$ is defined as:

$$p(y|\mathcal{I}) = p([MASK] = \mathcal{M}(y)|\mathcal{I}_{\mathcal{T}}) = \frac{\exp(w_{\mathcal{M}(y)} \cdot h_{[MASK]})}{\sum_{y' \in \mathcal{Y}} \exp(w_{\mathcal{M}(y')} \cdot h_{[MASK]})}, \quad (1)$$

where $\mathcal{I}_{\mathcal{T}} = \mathcal{T}(\mathcal{I})$ is the text obtained by applying the template $\mathcal{T}$ to the instance $\mathcal{I}$, resulting in a sentence with exactly one masked token ([MASK]). Here, $h_{[MASK]}$ represents the embedding of the [MASK] token, and w_v is the pre-softmax token embedding for the token $v \in \mathcal{V}$ in the PLM's vocabulary. The predicted label for the instance $\mathcal{I}$ is the label $y \in \mathcal{Y}$ with the highest predicted probability.

4 Methodology

This section introduces ColdSelect, a method for jointly selecting verbalizers and few-shot instances by effectively modeling data diversity. Section 4.1 outlines the problem, and Sect. 4.2 highlights the instance-verbalizer relationship in cold-start settings. Given a dataset $\mathcal{D}$ with N instances, ColdSelect begins by extracting embeddings of the PLM's vocabulary tokens and $h_{[MASK]}$ embeddings of dataset instances, mapping them into a shared embedding space. To capture the dataset's diversity, ColdSelect applies PCA for dimensionality reduction, followed by KMeans clustering and negative silhouette loss to produce robust, well-separated clusters. To refine these clusters, vocabulary-only clusters are discarded, and instances from instance-only clusters are reassigned to the nearest mixed clusters, ensuring that all clusters contain both instances and vocabulary tokens. Section 4.3 details the cluster creation process. The refined clusters are then passed to the Selection and Annotation module, which uses cohesion, separation, and impurity metrics to model cluster uncertainty and selects a subset of instances to obtain annotations and verbalizer tokens under the given budget $\mathcal{B}$. Section 4.4 provides details about the module, and Sect. 4.5 demonstrates the optimality of the proposed selection process. Figure 1 provides an overview of ColdSelect.

4.1 Problem Formulation

Given a PLM $\mathcal{L}$ trained with MLM objective, an unlabeled corpus $\mathcal{D}$ containing N instances, a template $\mathcal{T}$, and a labeling budget $\mathcal{B}$, the objective is to minimize uncertainty in classifying instances in $\mathcal{D}$ into predefined labels $\mathcal{Y}$ (binary or multi-class).

4.2 Relationship Between Instance and Verbalizer Selection

In prompt-based learning, the probability of assigning a label $y \in \mathcal{Y}$ to an input instance $\mathcal{I}$ is defined in Eq. (1). In the equation, the dot product $w_{\mathcal{M}(y)} \cdot h_{[MASK]}$ is maximized when the embeddings are similar or have high cosine similarity, assuming normalized embeddings:

$$\cos\_\mathrm{sim}(w_{\mathcal{M}(y)}, h_{[MASK]}) = \frac{w_{\mathcal{M}(y)} \cdot h_{[MASK]}}{\|w_{\mathcal{M}(y)}\| \|h_{[MASK]}\|}. \tag{2}$$

To ensure optimal classification, the verbalizer token $w_{\mathcal{M}(y)}$ must be close to the $h_{[MASK]}$ embedding of instances assigned to label y in the shared embedding space.

4.3 Modeling Data Diversity for Cluster Creation

Prompt-based learning utilizes PLMs to extract pre-softmax embeddings w_v for vocabulary tokens $v \in \mathcal{V}$ and $h_{[MASK]}$ embeddings for instances in the dataset $\mathcal{D}$. Since both types of embeddings are derived from the same PLM, they are mapped into a shared embedding space to enable meaningful comparisons. However, the high dimensionality of these embeddings can result in uniformly high cosine similarity values, diminishing their discriminative power. To address this, we first perform dimensionality reduction to enhance the separability of embeddings. This is followed by clustering and cluster optimization, ensuring the effective modeling of data diversity.

Dimensionality Reduction with PCA: To reduce the dimensionality of embeddings, we apply Principal Component Analysis (PCA) [38], which projects the embeddings into a lower-dimensional space while retaining most of the variance:

$$z = XW, \quad W = \arg\max_{W} \|XW\|_F, \quad \text{s.t. } W^\top W = I, \tag{3}$$

where X is the matrix of original embeddings, W is the transformation matrix, and z is the reduced embedding. After reduction, we normalize the embeddings to ensure cosine similarity effectively represents the dot product.

Clustering with KMeans: To cluster the embeddings based on their similarity, we use KMeans clustering [20], where the number of clusters is set to K. KMeans minimizes the within-cluster variance:

$$\mathcal{L}_{kmeans} = \sum_{k=1}^{K} \sum_{x \in C_k} \|x - \mu_k\|^2, \tag{4}$$

where C_k is the set of points in cluster k, and μ_k is the cluster centroid. Clustering is performed to group vocab tokens and instance embeddings to maximize intra-cluster similarity and facilitate the subsequent selection of verbalizers and instances.

Optimizing Clustering with Negative Silhouette Loss: Since KMeans is sensitive to initialization and may produce suboptimal clusters, we refine the clustering using negative silhouette loss [26], which measures cluster cohesion and separation. The silhouette score for a point i is defined as:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}, \tag{5}$$

where $a(i)$ is the average distance to other points in the same cluster, and $b(i)$ is the smallest average distance to points in any other cluster. The objective is to minimize the negative silhouette score:

$$\mathcal{L}_{sil} = -\frac{1}{N}\sum_{i=1}^{N} S(i), \tag{6}$$

where N is the total number of points. This optimization ensures clusters are well-separated and cohesive, improving the reliability of the clustering process.

Filtering and Cluster Refinement: After clustering, three types of clusters typically emerge: (1) *Mixed Clusters:* Contain both token and instance embeddings, (2) *Token-Only Clusters:* Contain only token embeddings, often representing outliers, and (3) *Instance-Only Clusters:* Contain only instance embeddings, typically distant from token distributions. We discard token-only clusters as outliers and reassign instances from instance-only clusters to the nearest mixed cluster based on cosine similarity to the centroid:

$$C_{assign} = \underset{C_k \in \mathcal{C}}{\arg\max}\, \text{cos\_sim}(\mu_k, h_{\mathcal{I}}), \tag{7}$$

where μ_k is the centroid of cluster $C_k \in \mathcal{C}$, and $h_{\mathcal{I}}$ is the $h_{[MASK]}$ embedding of instance $\mathcal{I}$. This refinement ensures all clusters are meaningful and suitable for verbalizer and instance selection.

By combining dimensionality reduction, clustering, optimization, and refinement, our approach ensures that the final clusters capture the diversity and dependency between PLM vocab token and instance embeddings, laying the foundation for robust instances for annotation and verbalizer selection.

4.4 Modeling Uncertainty for Cluster Selection and Annotation

The objective of ColdSelect is to minimize uncertainty in classifying instances in $\mathcal{D}$ into pre-defined labels $\mathcal{Y}$ (binary or multi-class). To achieve this, we leverage three key factors, cohesion, separation, and impurity, that collectively model cluster uncertainty. This framework ensures efficient use of the given labeling budget $\mathcal{B}$ in a cold-start setting. Below, we outline the motivation and mathematical formulation for each step in ColdSelect.

Cohesion: For dense clusters where embeddings are close to the cluster centroid μ_k, the probability that all instances belong to the same class is high. Cohesion models cluster density as:

$$\mathit{cohesion}(C_k) = \frac{1}{|C_k|} \sum_{x \in C_k} \cos\_\mathrm{sim}(x, \mu_k), \tag{8}$$

where $x \in C_k$ are the embeddings in cluster C_k, and $\cos\_\mathrm{sim}(x, \mu_k)$ measures their similarity to the cluster centroid.

Separation: Clusters far from others may represent outlier classes where all instances belong to the same class. Separation quantifies the distance between clusters as:

$$\mathit{separation}(C_k) = \max_{C_{k'} \neq C_k \in \mathcal{C}} \cos\_\mathrm{sim}(\mu_k, \mu_{k'}), \tag{9}$$

where $\mu_{k'}$ is the cluster centroid of cluster $C_{k'}$.

Impurity: Dense clusters may still contain instances from multiple classes, while sparse clusters can have low label diversity. Impurity models label diversity as:

$$\mathit{impurity}(C_k) = 1 - \frac{\max_{l \in \mathcal{L}} \mathit{count}_{C_k}(l)}{\mathit{total}(C_k)}, \tag{10}$$

where $\mathcal{L}$ is the set of labels, $\mathit{count}_{C_k}(l)$ is the number of instances with label l in cluster C_k, and $\mathit{total}(C_k)$ is the total number of instances in C_k.

Instance Classification Uncertainty Minimization. We model the uncertainty minimization problem as a cluster-level impurity minimization task. Since every instance belongs to a cluster, reducing impurity at the cluster level effectively minimizes uncertainty in instance classification. To achieve this, we prioritize clusters with high impurity at each step, operating on the principle that annotating instances within these clusters will significantly reduce their impurity. Additionally, we incorporate intra-cluster cohesion and inter-cluster separation to ensure the selection of representative clusters while avoiding redundancy. As a result, at each step T, the cluster that maximizes the following equation is selected for annotation.

$$\mathcal{C}_T = \arg\max_{C_k \in \mathcal{C}} \mathbb{E}\big[\mathit{cohesion}(C_k) + \mathit{separation}(C_k) + \mathit{impurity}(C_k)\big], \tag{11}$$

The inclusion of $\mathit{cohesion}(C_k)$ in the cluster selection process inherently favors dense clusters, which is beneficial, particularly under low labeling budgets.

Dense clusters, characterized by closely clustered embeddings, often indicate that instances belong to the same class, making them ideal for efficient labeling. Prioritizing these clusters ensures that each labeled instance has maximum impact, minimizing uncertainty while conserving resources. Additionally, this strategy reduces noise in the early stages, creating a strong foundation for subsequent labeling. Over time, the balance between cohesion, separation, and impurity ensures that sparse and diverse clusters are also addressed, leading to optimal resource allocation and improved model performance.

Initialization for Cold-Start Settings. In a cold-start scenario, where no labeled instances are initially available, the metrics are initialized as follows: *cohesion* is computed using Eq. (8), *separation* is determined using Eq. (9), and *impurity* is set to 0 for all groups.

Dynamic Updates for Cohesion and Separation. To incorporate instance labeling dynamically, we replace static cluster centroids with embeddings of selected verbalizer tokens. Verbalizer tokens provide a more contextually relevant reference for cluster evaluation, as class probabilities are determined by the dot product between the $h_{[MASK]}$ embedding of an instance and these tokens. Unlike fixed centroids, verbalizer tokens adapt as labels are assigned, capturing evolving cluster dynamics effectively. Eq. (8) and Eq. (9) are updated as follows:
Cohesion:

$$cohesion(C_k) = \frac{1}{|C_k|} \sum_{x \in C_k} \max_{v \in \mathcal{V} \& v \in C_k} \cos\_\mathrm{sim}(x, v), \tag{12}$$

where $v \in \mathcal{V}$ are verbalizer tokens in cluster C_k.
Separation:

$$separation(C_k) = \max_{v \in \mathcal{V} \& v \notin C_k} \cos\_\mathrm{sim}(\mu_k, v). \tag{13}$$

Labeling Policy. We categorize the selected cluster $\mathcal{C}_T$ as labeled if at least one instance in the cluster is already labeled. Otherwise, it is categorized as unlabeled. Depending on the categorization, the instance selection for labeling proceeds as follows:
If $\mathcal{C}_T$ is **unlabeled**, select the instance nearest to the cluster centroid μ for labeling:

$$\mathcal{I}_{select_T} = \arg\max_{x \in \mathcal{C}_T} \cos\_\mathrm{sim}(x, \mu). \tag{14}$$

Assign the label obtained for $\mathcal{I}_{select_T}$ to the nearest vocab token and add it to $\mathcal{V}$:

$$v_{select_T} = \arg\max_{v \in \mathcal{C}_T} \cos\_\mathrm{sim}(v, \mathcal{I}_{select_T}). \tag{15}$$

If $\mathcal{C}_T$ is **labeled**, select the instance farthest from already labeled instances and obtain verbalizer token following Eq. (15):

$$\mathcal{I}_{select_T} = \arg\max_{x \in \mathcal{C}_T} \min_{x \neq x' \in \mathrm{labeled}} \cos\_\mathrm{sim}(x, x'). \tag{16}$$

Stopping Criterion. The selection process continues until the labeling budget $\mathcal{B}$ is exhausted. By iteratively targeting clusters with maximum impurity, ColdSelect optimally selects instances and verbalizer tokens to reduce classification uncertainty and enhance model performance.

4.5 Optimal Selection Process

The proposed approach reduces classification uncertainty in $\mathcal{D}$ by integrating **cohesion**, **separation**, and **impurity** metrics into the cluster selection process. This balanced scoring function enables effective exploration and exploitation, focusing on dense, distinct clusters initially and gradually addressing sparse or diverse clusters as labels are acquired. Metrics are initialized based on embedding proximity and dynamically updated during labeling, allowing adaptability in cold-start settings. By combining these metrics, the approach optimally utilizes the labeling budget $\mathcal{B}$ to reduce uncertainty, avoid redundancy, and ensure robust classification for diverse datasets.

5 Experiments

In this section, we evaluate the performance of ColdSelect in reducing uncertainty in classifying instances on several benchmark datasets from diverse domains: SST-2 [33], MR [24], CR [9], Subj [25], CoLA [37], AG News [42], Yelp [22], and IMDB [19][1]. Table 1 provides a summary of the datasets, including their type, number of classes, and the templates used.

5.1 Evaluation Metrics

We use **Accuracy (Acc.)** as the primary evaluation metric across all datasets to measure the effectiveness of ColdSelect and baselines in reducing classification uncertainty.

Table 1. Statistics of the Datasets

Dataset	Type	$\lvert y \rvert$	Labels	#Test Instances	Template
SST-2	Sentiment Analysis	2	positive, negative	872	<S>. It was [MASK]
MR	Sentiment Analysis	2	positive, negative	2,000	<S>. It was [MASK]
CR	Sentiment Analysis	2	positive, negative	2,000	<S>. It was [MASK]
Subj	Subjectivity Classification	2	subjective, objective	2,000	<S>. It was [MASK]
CoLA	Acceptability Classification	2	grammatical, not grammatical	1,042	<S>. This is [MASK]
AG News	News Classification	4	world, sports, business, technology	7,600	[MASK] News: <S>
Yelp-full	Sentiment Analysis	5	very positive, positive, neutral, negative, very negative	38,000	<S>. It was [MASK]
IMDB	Sentiment Analysis	2	positive, negative	25,000	<S>. It was [MASK]

[1] The code is available at https://github.com/Mohna0310/COLDSELECT.

Table 2. Few-shot instance selection results on three datasets using RoBERTa-base with standard finetuning, following [39]. Accuracy is reported on the test set, with best and runner-up models highlighted in bold and underlined, respectively.

Dataset	|y|	$\|\mathcal{B}\|$	Random	Uncertainity	CAL	BERT-KM	Coreset	Margin-KM	ALPS	**TPC**	PATRON	Random-g	ColdSelect
IMDB	2	32	80.2	81.9	77.8	79.2	74.5	76.7	82.2	82.8	85.5	84.1	**87.37**
		64	82.6	84.7	81.2	84.9	82.8	84.0	86.1	84.0	87.3	85.3	**88.39**
		128	86.6	87.1	87.9	88.5	87.8	88.2	87.5	88.1	89.6	88.7	**90.61**
Yelp-full	5	32	30.2	32.7	36.6	35.2	32.9	32.7	36.8	32.6	35.9	34.1	**39.58**
		64	42.5	36.8	41.2	39.3	39.9	39.8	40.3	39.7	44.4	41.5	**46.72**
		128	47.7	41.3	45.7	46.4	49.4	47.1	45.1	46.8	51.2	48.9	**54.16**
AG News	4	32	73.7	73.7	69.4	79.1	78.6	75.1	78.4	80.7	83.2	81.2	**84.69**
		64	80.0	80.0	78.5	82.4	82.0	81.1	82.6	83.0	85.3	83.8	**87.16**
		128	84.5	82.5	81.3	85.6	85.2	85.7	84.3	85.7	87.0	86.1	**88.26**

5.2 Baseline Methods

Since ColdSelect is the first approach to jointly select both verbalizers and few-shot instances, we compare it with baselines for (1) few-shot instance selection and (2) verbalizer selection.

Few-Shot Instance Selection Baselines: **Random**: randomly selects samples for annotation. **Uncertainty** [30]: selects instances with the highest uncertainty post-calibration using entropy [13]. **CAL** [21]: uses Kullback-Leibler (KL) divergence to guide sample selection. **Coreset** [31]: minimizes the maximum Euclidean distance between a sample and its nearest cluster centroid. **BERT-KM** [4]: clusters embeddings using KMeans and selects samples closest to centroids. **Margin-KM** [23]: selects samples based on the margin between the two highest probabilities within clusters. **ALPS** [40]: uses BERT's MLM loss to compute surprisal embeddings for sample selection. **TPC** [7]: selects instances with the highest density in each cluster. **PATRON** [39]: employs a partition-then-rewrite strategy to enhance sample diversity. **Random-g**: randomly selects refined clusters at each step, bypassing the proposed selection process while adhering to the labeling policy.

Automatic Verbalizer Selection Baselines: **LM-BFF** [6]: uses T5 to automatically generate verbalizers with few-shot examples. **ProtoVerb** [5]: constructs prototype-based verbalizers using contrastive learning. For verbalizer selection, we also compare ColdSelect against manual verbalizers created by humans.

5.3 Experimental Settings

We conduct experiments using RoBERTa-base (125M parameters) and RoBERTa-large (355M parameters) PLMs, following the setups of [39] and [6], respectively. For KMeans clustering, we set the random seed to 42 and the number of clusters to 40 and performed cluster optimization over five iterations.

Table 3. Few-shot instance selection results on three benchmark datasets using RoBERTa-base, following [39]. Prompt-based finetuning is performed, with accuracy reported on the test set. Best and runner-up models are highlighted in bold and underlined, respectively.

Dataset	|y|	$\|\mathcal{B}\|$	Random	Uncertainity	CAL	BERT-KM	Coreset	Margin-KM	ALPS	TPC	PATRON	Random-g	ColdSelect
IMDB	2	32	81.8	82.4	79.6	81.7	85.5	86.0	83.5	84.5	86.5	85.2	**89.48**
		64	85.6	86.0	81.1	84.2	87.8	87.6	84.4	85.8	88.8	87.2	**91.42**
		128	87.7	88.4	83.0	88.5	88.9	89.1	88.9	88.0	89.3	88.5	**91.23**
Yelp-full	5	32	48.9	46.6	47.9	45.5	46.0	47.5	47.0	49.8	50.5	49.1	**53.31**
		64	51.0	49.9	49.4	51.9	48.8	52.6	52.8	52.3	53.6	51.5	**56.29**
		128	51.3	50.8	48.7	51.5	53.7	54.2	51.7	51.0	55.6	53.2	**57.16**
AG News	4	32	83.1	82.8	81.4	84.9	85.1	84.6	84.2	85.6	86.8	85.7	**87.23**
		64	84.5	84.3	82.6	86.5	86.4	85.9	86.2	85.6	87.4	86.8	**88.18**
		128	84.9	83.1	83.0	87.6	87.5	87.1	87.5	87.0	87.8	87.4	**89.95**

Table 4. Performance comparison of ColdSelect with Random, Random-g, and other methods on five benchmark datasets using RoBERTa-large LLM, following [6]. Experiments use fixed manual templates with $K = 16$ few-shot instances per class to obtain automatic verbalizers. The maximum budget $\mathcal{B}$ required by Random, Random-g, and ColdSelect are 66, 52, and 44, respectively, showing ColdSelect's efficiency in balancing class labels and optimizing the labeling budget. Accuracy (%) is reported on test set, with the best results highlighted in bold.

Dataset	Fine-tuning			Prompt-based FT			LM-BFF			ProtoVerb		
	Rand	Rand-g	ColdSelect	Rand	Rand-g	ColdSelect	Rand	Rand-g	ColdSelect	Rand	Rand-g	ColdSelect
SST-2	81.4	83.0	**87.5**	92.7	93.1	**93.0**	92.6	92.8	**93.2**	86.9	87.0	**87.1**
MR	76.9	77.8	**79.5**	87.0	87.5	**88.9**	86.6	87.0	**87.8**	60.0	62.4	**66.2**
CR	75.8	76.3	**78.0**	90.3	90.7	**91.5**	90.2	90.6	**91.7**	68.7	70.1	**76.8**
Subj	90.8	91.2	**93.2**	91.2	91.5	90.2	92.3	92.5	**93.7**	75.7	76.0	**76.3**
CoLA	72.4	73.2	**73.8**	52.9	56.2	**58.3**	52.9	53.6	**54.5**	53.9	55.3	**58.9**

5.4 Results and Discussion

The results in Tables 2 and 3 demonstrate ColdSelect's superior performance over PATRON and other baselines by effectively leveraging data diversity and label uncertainty. Unlike PATRON, which relies on prompt-based uncertainty propagation and a static partition-then-rewrite (PTR) strategy, ColdSelect dynamically updates cluster centroids with verbalizer token embeddings, ensuring better alignment with evolving label distributions and reducing noise from outliers. By integrating cohesion, separation, and impurity metrics, ColdSelect balances exploration and exploitation, enabling early-stage labeling of dense clusters while progressively addressing sparse or diverse ones. This adaptability allows ColdSelect to surpass PATRON across various datasets, including IMDB in Table 2, where it achieves 87.37% accuracy at $|\mathcal{B}| = 32$ (1.9% higher than PATRON), Yelp-full in Table 2, where it achieves largest accuracy gain over PATRON, indicating its robustness in handling highly imbalanced class distributions, and AG News in Table 3, where it scales effectively to 89.95% at $|\mathcal{B}| = 128$.

These results highlight ColdSelect's ability to optimize labeling budgets and handle complex, multi-class distributions more effectively than existing methods.

The results in Table 4 demonstrate ColdSelect's effectiveness across all datasets and experimental setups, consistently outperforming Random and Random-g. For instance, in the Fine-tuning setting, ColdSelect achieves 87.5% on SST-2, surpassing Random (81.4%) and Random-g (83.0%). Similarly, on MR and CR, ColdSelect outperforms both baselines, achieving 79.5% and 78.0%, respectively. In the Prompt-based FT setup, ColdSelect achieves 91.5% on CR and 58.3% on CoLA, significantly higher than Random and Random-g. ColdSelect's reduced labeling budget ($\mathcal{B} = 44$) compared to Random ($\mathcal{B} = 66$) and Random-g ($\mathcal{B} = 52$) highlights its efficiency in selecting diverse, representative instances. By ensuring that each labeled instance maximally reduces uncertainty, ColdSelect optimally utilizes limited annotation budgets, validating its ability to optimize instance selection while improving performance across binary classification tasks. ColdSelect shows greater improvements in datasets with complex class distributions, such as Yelp-full and AG News, compared to binary tasks like SST-2. The results suggest that clustering-based selection of ColdSelect is particularly effective in capturing fine-grained distinctions between similar classes.

The ablation study in Table 5 further supports this by showing that removing impurity modeling leads to a drop in accuracy across IMDB, AG News, and Yelp-full datasets, reinforcing the importance of selecting diverse and representative instances. Using cohesion alone provides a baseline improvement by prioritizing dense clusters (e.g., 80.50% on IMDB), but it lacks robustness. Adding separation enhances performance (e.g., 82.20% on IMDB) by ensuring inter-cluster distinctiveness, reducing redundancy, and better capturing outlier classes. Including impurity with cohesion offers slight gains (e.g., 81.40% on IMDB) by addressing label diversity within clusters. However, the full combination of all three metrics yields the highest accuracy across datasets (e.g., 87.37% on IMDB), demonstrating their complementary roles in capturing cluster density, distinctiveness, and label diversity to optimize labeling budgets effectively.

Table 5. Ablation study on ColdSelect's performance with $\mathcal{B} = 32$ using RoBERTa-base and standard fine-tuning, following [39]. Accuracy (%) on the test set is reported, with best and runner-up results highlighted in bold and underlined.

Model Variant	Cohesion	Separation	Impurity	IMDB	AG News	Yelp-full
Only Cohesion	✓			80.50	78.30	30.80
Cohesion + Separation	✓	✓		<u>82.20</u>	<u>80.10</u>	<u>33.70</u>
Cohesion + Impurity	✓		✓	81.40	79.00	32.90
Cohesion + Separation + Impurity (ColdSelect)	✓	✓	✓	**87.37**	**84.69**	**39.58**

6 Conclusion

In this study, we proposed ColdSelect, a novel approach, for instance, and verbalizer selection in prompt-based learning, explicitly modeling data diversity and label uncertainty. By integrating cohesion, separation, and impurity metrics, ColdSelect effectively identifies representative instances and optimizes labeling budgets. Extensive experiments on benchmark datasets demonstrate that ColdSelect consistently outperforms state-of-the-art methods, achieving robust performance in both standard and prompt-based fine-tuning. These results validate the importance of modeling intra-cluster density, inter-cluster distinctiveness, and label diversity, making ColdSelect a powerful solution for challenging cold-start scenarios and enhancing generalization in prompt-based classification tasks.

The effectiveness of ColdSelect in optimizing annotation budgets makes it highly applicable in low-resource NLP settings, active learning scenarios, and domain adaptation tasks. By reducing redundancy in labeled instances and improving generalization, this method can enhance model performance in real-world applications such as customer sentiment analysis, fake news detection, and biomedical text classification. Moreover, its ability to optimize instance selection without extensive labeled data makes it particularly relevant for emerging fields like legal text classification, financial risk assessment, and cross-lingual NLP, where annotated data is often scarce. Future extensions of ColdSelect could explore its adaptability to multilingual and multimodal datasets, further broadening its impact across diverse AI applications.

7 Limitations and Future Work

While ColdSelect excels in data-diverse scenarios, its reliance on PCA-based dimensionality reduction may introduce biases, as it assumes linear separability. Future work could explore adaptive non-linear transformations like kernel PCA or deep representation learning to enhance cluster separability. Additionally, the fixed labeling budget may not suit all datasets; a dynamic allocation strategy based on entropy minimization and cluster impurity could improve efficiency.

ColdSelect also depends on KMeans clustering, which may not always capture complex structures in high-dimensional spaces. Alternative clustering methods like hierarchical clustering or DBSCAN could enhance robustness. Additionally, integrating reinforcement learning could further refine instance selection by dynamically adapting to dataset characteristics.

Acknowledgement. The work is supported by the US National Science Foundation under grant NSF-CAREER 2237831.

References

1. Achiam, J., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Chakraborty, M., Kulkarni, A., Li, Q.: Open-domain aspect-opinion co-mining with double-layer span extraction. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22, pp. 66–75. Association for Computing Machinery, New York (2022). https://doi.org/10.1145/3534678.3539386
3. Chakraborty, M., Kulkarni, A., Li, Q.: Zero-shot approach to overcome perturbation sensitivity of prompts. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 5698–5711 (2023)
4. Chang, E., Shen, X., Yeh, H.S., Demberg, V.: On training instance selection for few-shot neural text generation. In: Proceedings of the 59th Annual Meeting of the ACL and the 11th IJCNLP, vol. 2: Short Papers, pp. 8–13 (2021)
5. Cui, G., Hu, S., Ding, N., Huang, L., Liu, Z.: Prototypical verbalizer for prompt-based few-shot tuning. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 7014–7024 (2022)
6. Gao, T., Fisch, A., Chen, D.: Making pre-trained language models better few-shot learners. In: Proceedings of the 59th Annual Meeting of the ACL and the 11th IJCNLP, vol. 1: Long Papers, pp. 3816–3830 (2021)
7. Hacohen, G., Dekel, A., Weinshall, D.: Active learning on a budget: opposite strategies suit high and low budgets. In: International Conference on Machine Learning, pp. 8175–8195. PMLR (2022)
8. Hambardzumyan, K., Khachatrian, H., May, J.: Warp: word-level adversarial reprogramming. In: Proceedings of the 59th Annual Meeting of the ACL and the 11th IJCNLP, vol. 1: Long Papers, pp. 4921–4933 (2021)
9. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177 (2004)
10. Hu, S., Ding, N., et al.: Knowledgeable prompt-tuning: incorporating knowledge into prompt verbalizer for text classification. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 2225–2240 (2022)
11. Kulkarni, A., Chakraborty, M., Xie, S., Li, Q.: Optimal budget allocation for crowdsourcing labels for graphs. In: Evans, R.J., Shpitser, I. (eds.) Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence. Proceedings of Machine Learning Research, vol. 216, pp. 1154–1163. PMLR (2023)
12. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2021)
13. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 3–12 (1994)
14. Li, X.L., Liang, P.: Prefix-tuning: optimizing continuous prompts for generation. In: Proceedings of the 59th Annual Meeting of the ACL and the 11th IJCNLP, vol. 1: Long Papers, pp. 4582–4597 (2021)
15. Liu, C., Wang, H., Xi, N., Zhao, S., Qin, B.: Global prompt cell: a portable control module for effective prompt tuning. In: CCF International Conference on Natural Language Processing and Chinese Computing, pp. 657–668 (2023)

16. Liu, J., Shen, D., et al.: What makes good in-context examples for gpt-3? In: Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pp. 100–114 (2022)
17. Liu, X., Zheng, Y., Du, Z., et al.: Gpt understands, too. AI Open (2023)
18. Liu, Y., Ott, M., et al.: Roberta: a robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
19. Maas, A., Daly, R.E., et al.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150 (2011)
20. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press (1967)
21. Margatina, K., Vernikos, G., Barrault, L., Aletras, N.: Active learning by acquiring contrastive examples. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 650–663 (2021)
22. Meng, Y., Shen, J., Zhang, C., Han, J.: Weakly-supervised hierarchical text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6826–6833 (2019)
23. Müller, T., Pérez-Torró, G., Basile, A., et al.: Active few-shot learning with fasl. In: International Conference on Applications of Natural Language to Information Systems, pp. 98–110 (2022)
24. PANG, B.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 2002 (2002)
25. PANG, B.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL) 2004 (2004)
26. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
27. Schick, T., Schmid, H., Schütze, H.: Automatically identifying words that can serve as labels for few-shot text classification. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 5569–5578 (2020)
28. Schick, T., Schütze, H.: 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, pp. 255–269 (2021)
29. Schick, T., Schütze, H.: It's not just size that matters: small language models are also few-shot learners. In: Proceedings of the 2021 NAACL-HLT, pp. 2339–2352 (2021)
30. Schröder, C., Niekler, A., Potthast, M.: Revisiting uncertainty-based query strategies for active learning with transformers. In: Findings of the ACL: ACL 2022, pp. 2194–2203 (2022)
31. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. In: International Conference on Learning Representations (2018)
32. Shin, T., Razeghi, Y., et al.: Autoprompt: eliciting knowledge from language models with automatically generated prompts. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4222–4235 (2020)

33. Socher, R., Perelygin, A., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)
34. Su, H., Kasai, J., et al.: Selective annotation makes language models better few-shot learners. arXiv preprint arXiv:2209.01975 (2022)
35. Wang, H., Liu, C., et al.: Prompt combines paraphrase: teaching pre-trained models to understand rare biomedical words. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 1422–1431 (2022)
36. Wang, H., Zhao, S., et al.: Manifold-based verbalizer space re-embedding for tuning-free prompt-based classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 19126–19134 (2024)
37. Warstadt, A., Singh, A., Bowman, S.R.: Neural network acceptability judgments. Trans. Assoc. Comput. Linguist. **7**, 625–641 (2019)
38. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemom. Intell. Lab. Syst. **2**(1–3), 37–52 (1987)
39. Yu, Y., Zhang, R., et al.: Cold-start data selection for better few-shot language model fine-tuning: a prompt-based uncertainty propagation approach. In: Proceedings of the 61st Annual Meeting of the ACL, vol. 1: Long Papers, pp. 2499–2521 (2023)
40. Yuan, M., Lin, H.T., Boyd-Graber, J.: Cold-start active learning through self-supervised language modeling. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 7935–7948 (2020)
41. Zhang, N., Li, L., et al.: Differentiable prompt makes pre-trained language models better few-shot learners. In: International Conference on Learning Representations (2021)
42. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. Adv. Neural Inf. Process. Syst. **28** (2015)