CLL-RetICL: Contrastive Linguistic Label Retrieval-based In-Context Learning for Text Classification via Large Language Models

Chaohao Lin[1], Kaida Wu[1], Peihao Xiang[1], Yanzhao Wu[1], Ou Bai[1],
[1]Florida International University, Miami, USA
Correspondence: obai@fiu.edu

Abstract

Recent research has delved into Retrieval-based In-Context Learning (RetICL), leveraging the power of large language models (LLMs) for text classification. Despite its promise, a persistent challenge lies in effectively retrieving relevant demonstrations from a support set. Many existing approaches have overlooked the essential role of linguistic label information in guiding this retrieval process. To bridge this gap, we present Contrastive Linguistic Label Retrieval-based In-Context Learning (CLL-RetICL), a novel framework designed to identify the most relevant and impactful sentences without altering the model parameters. Our approach uniquely integrates sentence-query similarity with sentence-label similarity, enabling a more nuanced and comprehensive evaluation of relevance. We tested CLL-RetICL across diverse text classification tasks and evaluated its performance on various LLMs. Experimental results demonstrate that CLL-RetICL consistently outperforms previous retrieval methods that do not incorporate linguistic label information. These findings highlight the critical importance of linguistic label-aware selection in enhancing text classification accuracy.[1]

1 Introduction

A linguistic label represents the semantics of a category and plays a vital role in text classification tasks. Human annotators rely on the meaning conveyed by these labels to accurately categorize text. Depending on the specific requirements of a custom classification task, a linguistic label can often be substituted with synonyms or more descriptive phrases to better align with the task's context.

Recently, researchers have begun exploring few-shot in-context learning (ICL) using LLMs for text classification tasks (Luo et al., 2024; Yu et al., 2023; Chae and Davidson, 2023; Rouzegar and Makrehchi, 2024). Instead of selecting static, predefined demonstration sets for ICL, RetICL adopts a dynamic, context-sensitive approach (Zhao et al., 2021; Lu et al., 2022; Xu et al., 2024; Wu et al., 2023). At its core, adaptive demonstration selection leverages a specialized retriever to intelligently curate tailored demonstrations for each task input. RetICL has gained popularity because prior research suggests that context-insensitive demonstrations can limit the full potential of LLMs (Luo et al., 2024; Wu et al., 2022). Despite RetICL consistently surpassing approaches based on random or static demonstrations, it still remains an open challenge to retrieve relevant demonstrations.

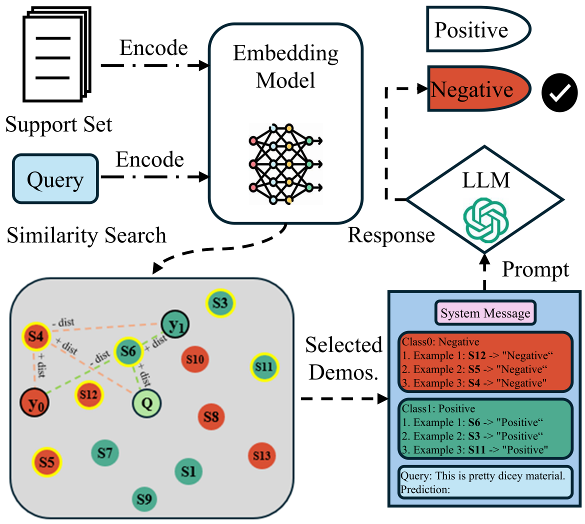
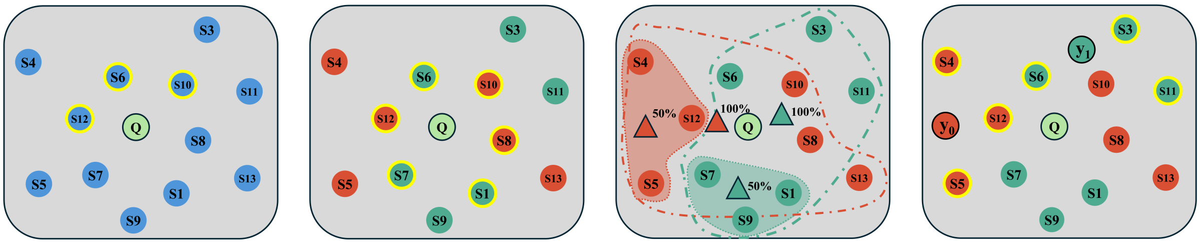


Figure 1: An illustration of CLL-RetICL with $N = 2$ and $k = 3$, demonstrating a prediction between Positive and Negative classes. Here, y_0 and y_1 represent the vector representations of the linguistic labels "Negative" and "Positive", respectively, in a pre-trained sentence embedding model. Similarly, $s_0, s_1, \ldots$ represent the vector representations of the sentences in a support set within the same pre-trained sentence embedding model.

To address the problem, previous researchers have proposed various strategies, including k-nearest neighbors (KNN), NwayKshot, and

[1]Our code is available: https://github.com/Toby28/CLLRetICL.

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1575–1590



(a) KNN (b) NwayKshot (c) Clustering-Based (d) Ours

Figure 2: A comparison of four different approaches to RetICL strategies. (a) KNN suffers from two key weaknesses: the copying effect and misleading by similarity. (b) NwayKshot always ignores any linguistic cues conveyed through the labels. (c) The performance of clustering-based approaches is constrained by challenges in accurately estimating category centers, typically denoted by triangles, as well as by the omission of query similarity considerations. (d) Our method avoids the copying effect, prevents misleading similarity, incorporates linguistic label information, utilizes fixed label category centers, and integrates query similarity.

clustering-based RetICL (Li et al., 2024; Pecher et al., 2024; Zhang et al., 2022a). However, these methods suffer from various challenges, as shown in Figure 2. To identify the most effective demonstrations, we analyzed failure cases. Our investigation revealed that there always exists a specific combination of demonstrations that enables LLMs to classify accurately. Additionally, our analysis uncovered that failure cases are error-prone: they often lie closer to the representation of an opposing linguistic label or near the representation of an incorrect label cluster center, despite their similarity to the query. In contrast, when the demonstrations are correctly combined, they align more closely with the representation of the intended linguistic label. A detailed discussion of these findings is presented in Section 3.

Building on these observations, we present a novel RetICL framework, CLL-RetICL (Contrastive Linguistic Label Retrieval-based In-Context Learning) as illustrated in Figure 1. Our approach introduces a trade-off method that computes a relevance score by integrating both sentence–query and sentence–label similarities, thereby effectively leveraging label information. Furthermore, to optimize the effectiveness of CLL-RetICL, we developed a universal *N-way K-shot* prompt structure applicable to all text classification tasks. This prompt design mitigates the copying effect and prevents LLMs from being misled by overly similar examples. Moreover, we demonstrate that the sentence embeddings of linguistic labels can serve as clustering centers—generated by a pre-trained sentence embedding model—to address the challenge of estimating clustering centers. Additionally, we initiate four variations for integrating the linguistic label style into RetICL and evaluate their effectiveness on four text classification datasets. Finally, to assess the generalizability of CLL-RetICL, we conduct experiments using Gemini (Team et al., 2024), Llama (Dubey et al., 2024), and Mistral (Jiang et al., 2024). Empirical experiments show that CLL-RetICL consistently outperforms both previous RetICL baselines and other variants across multiple datasets and LLMs. Ablation studies further reveal several key findings: (1) Effectiveness across variations: CLL-RetICL maintains strong performance across different k-shot settings, various pre-trained sentence embedding models, and multiple similarity functions. (2) Component dependency: The proposed method relies on the original component responsible for calculating sentence-query similarity; omitting this component degrades performance. (3) Impact of hyperparameters: Trade-off hyperparameters have a minor influence on the final classification accuracy. The following summarizes our main contributions:

- We present a novel perspective in which sentence embeddings of linguistic labels serve as highly accurate clustering centers, free from the biases introduced by limited support data and independent of data-driven constraints.

- We propose an innovative method, CLL-RetICL, which employs a rigorous relevance scoring metric that leverages linguistic label information to select high-quality demonstrations for improving LLMs in text classification tasks. Our approach does not require fine-tuning the pre-trained weights of either the sentence embedding models or LLMs.

- We conduct extensive experiments to evaluate the proposed method, achieving better performance on most datasets compared to existing RetICL methods.

2 Related Work

Text Classification via LLMs. Text classification via LLMs has recently demonstrated exceptional generalizability and reasoning capabilities, attracting significant research interest in their application to text classification tasks (Zhang et al., 2024; Wang et al., 2024; Fields et al., 2024). Existing methods can be broadly divided into two groups, depending on whether they involve adapting the parameters of LLMs or not. The first group concentrates on fine-tuning the parameters of LLMs to excel in custom text classification tasks (Chae and Davidson, 2023; Zhang et al., 2024; Yu et al., 2023; Jin et al., 2023). However, this approach generally demands significant computational resources to load the full LLM model parameters, and fine-tuning these models can often diminish their generalizability. The other category is known as ICL, or prompt engineering (Guo et al., 2024; Luo et al., 2024; Fan et al., 2024). While this method avoids the need to update LLM model parameters, it heavily depends on well-designed prompts, making it challenging to guide LLMs to consistently meet human expectations (Shi et al., 2023; Mavromatis et al., 2023; Edwards and Camacho-Collados, 2024).

RetICL. RetICL can generally be divided into two categories: approaches that retrain or fine-tune a retriever for specific text classification tasks, and approaches that utilize pre-trained language models without additional fine-tuning. An intuitive strategy for RetICL involves directly selecting a few similar sentences, leveraging readily available demonstration retrievers like those based on sentence embeddings. Existing methods include KATE (Liu et al., 2021), Z-ICL (Lyu et al., 2022) and ICL-ML (Milios et al., 2023). However, recent research has shown that selecting the most similar demonstrations can lead to the copying effect and misleading by similarity, degrading performance in text classification tasks (Olsson et al., 2022; Zhang et al., 2022b). To mitigate the issue of homogeneity in retrieval, clustering retrieval approaches ensure the selection of a diverse and representative set of demonstrations, which is critical to its effectiveness (Luo et al., 2024). Several methods exist, including

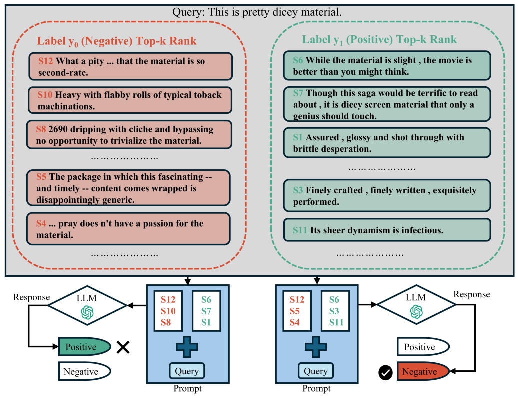


Figure 3: A comparison of the correct and incorrect demonstration combinations is presented. On the left, NwayKshot retrieves the top-*k* sentences most similar to the query from each group; however, this approach fails to classify the query correctly. In contrast, on the right, CLL-RetICL does not rely solely on proximity to the query, resulting in an accurate classification.

NwayKshot (Li et al., 2024), Votek (Su et al., 2022) and D-CALM (Hassan and Alikhani, 2023). While these approaches leverage label information and offer improvements, accurately estimating the clustering center for each category remains challenging. This difficulty arises because clustering center estimation is a data-driven process that depends on a support set.

The second category of RetICL involves fine-tuning or retraining a retriever model to rank relevant sentences using either in-domain or out-of-domain datasets for text classification tasks. There are established methods, such as PEFT (Tunstall et al., 2022), UDR (Li et al., 2023) and Ambig-ICL (Gao et al., 2023). These methods utilize label information and feedback to optimize model parameters, highlighting the essential role of labeled data in yielding valuable insights for text classification tasks. However, they often demand substantial computational resources and considerable time to construct a retriever.

3 Linguistic Label Retrieval Hypothesis

Previous studies have shown that retrieving sentences closest to the query and applying a clustering-based selection method can enhance the diversity of demonstrations while mitigating the risk of misleading results due to similarity (Li et al., 2020; Luo et al., 2024). Therefore, a question arises: are the clustering centers reliable? To explore this further, we analyze the distribution of clustering centers, as shown in Appendix C. Varying the proportion of fully supported data from 10% to 100% reveals that the distribution of clustering centers shifts according to the number of sentences in the support set. Notably, negative-labeled clustering centers tend to be less distinct within a certain range compared to positive-labeled ones. These findings suggest that clustering center estimation is inherently data-driven and prone to bias, making it difficult to accurately identify true clustering centers. On the other hand, by analyzing failure cases, we find that, for a given query, there is an optimal combination of demonstrations that can effectively guide LLMs to classify the query correctly. However, relying solely on the top-ranked closest demonstrations retrieved does not always yield accurate results. An example of this limitation is illustrated in Figure 3. To further investigate, we compared cases where the top-*k* closest demonstrations led to incorrect results versus cases where randomly selected demonstrations produced correct outcomes. We provide five examples of such instances in Appendix C. We found that incorrect nearest-neighbor demonstrations exhibit an error-prone tendency, being either closer to the linguistic representation of an opposite label, closer to the center of an incorrect label cluster, or both—despite being similar to the query. Conversely, in correct combinations, the selected demonstrations exhibit a stronger alignment with the correct tendency. For example, sentences with a Negative label tend to show higher similarity to the linguistic word "Negative" and the same holds for "Positive" label. Although correct demonstrations align closely with their respective cluster centers, we observe exceptions where a correct output contains sentences that are nearer to the center of an incorrect label cluster. Furthermore, even sentences closest to their correct cluster centers can still lead to classification errors due to inaccurate estimation of those centers.

Based on these observations, we hypothesize that the vector representations of linguistic labels should be explicitly incorporated into the retrieval process rather than relying on cluster center estimation. Compared to traditional clustering center estimation, this approach offers two advantages: (1) Independence from data Bias – The linguistic label clustering center is not data-driven, preventing bias introduced by the support set. (2) Leveraging linguistic information – Linguistic labels play a crucial role in zero-shot ICL, as LLMs rely entirely on these labels for text classification tasks.

4 Our Method: CLL-RetICL

Preliminary. Let the query set Q represent a task, where $q \in Q$ denotes a sample query for which we aim to find an answer via an LLM. In the context of RetICL, multiple demonstrations $(d_1, \ldots, d_k)$ are retrieved from a support set C. Each demonstration d_i consists of a sentence and its label, $(s_i, y_i) \in C$, where y_i belongs to the label set Y.

Overview. We present CLL-RetICL, a novel RetICL approach leveraging information extraction between demonstrations and linguistic labels to predict the correct label for a given query input q_i (Wang et al., 2023). Unlike earlier methods (Liu et al., 2021; Su et al., 2022; Li et al., 2022; Milios et al., 2023) that create input-label pairs by retrieving sentences closest to a given query, CLL-RetICL selects demonstrations that balance a trade-off by augmenting the corresponding label while penalizing others.

CLL-RetICL involves three key steps, as illustrated in Figure 1: (1) Retrieving more relevant sentences by integrating sentence-query similarity with sentence-label similarity (detailed in Section 4.1), (2) Forming demonstrations by organizing the retrieved demonstrations into an N-way K-shot format (discussed in Section 4.2), and (3) Making inferences through ICL (explained in Section 4.3).

4.1 Linguistic Label Retriever

RetICL employs a retrieval mechanism to identify k examples from C that are most relevant to a given query q. This process is guided by a similarity function, sim, which quantifies the relationship between a sentence s_i and a query q. The corresponding formula is as follows:

$$score_{RetICL} = sim(q, s_i) \tag{1}$$

To build on this hypothesis, CLL-RetICL incorporates sentence-query similarity with sentence-label similarity. Rather than solely considering the similarity distance between a sentence s_i and the query q, CLL-RetICL employs the following formula:

$$\begin{aligned} score_{CLL\text{-}RetICL} &= sim(q, s_i) \\ &+ w_1 * log \frac{\exp^{sim(s_i, y_i)}}{\frac{1}{n-1}\sum_{y \in Y}^{y \neq y_i} \exp^{sim(s_i, y)}} \end{aligned} \tag{2}$$

where w_1 is a trade-off hyperparameter that balances the relative importance of the corresponding terms in the objective function.

CLL-RetICL considers the relationship between sentences and linguistic labels by utilizing a similarity function. It increases the score based on the similarity between a sentence and its assigned correct label (referred to as the positive label) while decreasing the score based on the similarity between the sentence and other labels (referred to as negative labels). Additionally, we propose several variations and evaluate their performance through experiments. These include Positive Label Augment (PLA), Negative Label Penalty (NLP), and Contrastive Label (CTL). The corresponding formulas are provided below:

$$score_{PLA} = sim(q, s_i) + w_1 * sim(s_i, y_i) \quad (3)$$

$$score_{NLP} = sim(q, s_i) - w_1 * \frac{1}{n-1} \sum_{y \in Y}^{y \neq y_i} sim(s_i, y) \quad (4)$$

$$\begin{aligned} score_{CTL} =& sim(q, s_i) + w_1 * sim(s_i, y_i) \\ &- w_2 * \frac{1}{n-1} \sum_{y \in Y}^{y \neq y_i} sim(s_i, y) \end{aligned} \quad (5)$$

where w_1 and w_2 are trade-off hyperparameters.

Our methods ensure that the selected sentences (1) maintain a safe distance from q to prevent the copying effect (Olsson et al., 2022; Zhang et al., 2022b), (2) incorporate the information between sentences and linguistic labels and (3) align closely with the requirements of the custom text classification task.

4.2 *N-way K-shot*

We adopt a clustering-based retrieval method, as prior research suggests that *N-way K-shot* effectively addresses the issue of homogeneity (Li and Qiu, 2023). Here, we partition all sentences into N sub-groups, aiming to cluster sentences that share the same label. Our retriever selects top K high demonstrations according to above score formula from each sub-group, resulting in a final set of $N \times K$ demonstrations.

4.3 Inference

Finally, CLL-RetICL constructs a prompt by concatenating *N-way K-shot* input-label pairs $(s_1, y_1), (s_2, y_2), \ldots, (s_k, y_k)$ for each N-way label, along with the query input q. This prompt is then fed into a LLM, which generates a prediction using $argmax_{y \in Y} P(y|prompt)$. The universal prompt template for each text classification task is outlined in Table 7 in Appendix B.

5 Experimental Analysis

5.1 Experimental Setup

We evaluate multiple LLMs to identify factors affecting classification accuracy across four tasks. Key results are summarized in the main text, with additional details presented in the Appendix D.

5.1.1 Datasets

We conduct experiments on four widely recognized text classification tasks: SST2 (Socher et al., 2013), CoLA (Warstadt et al., 2018), CARER (Saravia et al., 2018) and BBCnews (Greene and Cunningham, 2006). Similar to conventional text classification methodologies, we treat the training sets as support sets and the test sets as query sets, while disregarding development sets if they exist. The detailed data statistics are provided in Appendix A and summarized in Table 5.

5.1.2 Baselines

We compare CLL-RetICL with the zero-shot approach as well as various RetICL methods.

Zero-shot predicts $argmax_{y \in Y} P(y|q)$ without using any demonstrations (Radford et al., 2019; Brown et al., 2020). This method utilizes LLMs and linguistic label information to enhance text classification.

Z-ICL leverages physical neighbors to avoid selecting demonstrations that are overly similar to the query. Furthermore, it introduces the use of synonymous labels to mitigate the copying effect, highlighting the potential for effectively utilizing the linguistic meaning of labels (Lyu et al., 2022).

KATE employs a standard KNN approach to retrieve demonstrations, which remains the most widely used method in RetICL (Liu et al., 2021).

NwayKshot is a clustering-based retrieval method designed to tackle the challenge of homogeneity in demonstrations (Li et al., 2024).

Cluster-TopN builds on NwayKshot but applies k-means clustering to identify the cluster centers. It then selects the demonstration closest to the center from each sub-group (Zhdanov, 2019; Hassan and Alikhani, 2023).

Votek selects k representatives from N sub-groups through a voting mechanism to best represent the group (Su et al., 2022).

LLM	Zero-shot		Z-ICL		KATE		Cluster-TopN		Votek		Nwaykshot		CLL-RetICL	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
SST2														
Gemini	$93.29_{.56}$	$.933_{.002}$	$92.31_{.29}$	$.923_{.005}$	$94.17_{.42}$	$.941_{.004}$	$\underline{94.93_{.47}}$	$\underline{.950_{.003}}$	$94.16_{.33}$	$.942_{.004}$	$94.67_{.47}$	$.947_{.002}$	$\mathbf{95.17_{.37}}$	$\mathbf{.952_{.004}}$
Llama	$94.83_{.62}$	$.948_{.004}$	$\mathbf{96.21_{.54}}$	$\mathbf{.962_{.006}}$	$94.78_{.56}$	$.948_{.004}$	$93.61_{.63}$	$.936_{.005}$	$94.77_{.47}$	$.948_{.004}$	$90.82_{.71}$	$.908_{.003}$	$\underline{95.06_{.43}}$	$\underline{.951_{.004}}$
Mistral	$90.08_{.32}$	$.901_{.002}$	$90.72_{.51}$	$.906_{.003}$	$93.78_{.21}$	$.938_{.003}$	$\underline{94.88_{.37}}$	$\underline{.949_{.004}}$	$94.34_{.46}$	$.943_{.002}$	$94.34_{.29}$	$.943_{.003}$	$\mathbf{95.60_{.21}}$	$\mathbf{.956_{.003}}$
Avg.	92.73	.927	93.08	.930	94.24	.942	$\underline{94.47}$	$\underline{.945}$	94.42	.944	93.27	.933	**95.28**	**.953**
CoLA														
Gemini	$68.26_{.56}$	$.663_{.008}$	$60.21_{.67}$	$.583_{.007}$	$70.08_{.84}$	$.641_{.008}$	$80.32_{.49}$	$.765_{.005}$	$81.43_{.63}$	$.783_{.007}$	$\underline{82.74_{.72}}$	$\underline{.795_{.008}}$	$\mathbf{83.60_{.91}}$	$\mathbf{.801_{.006}}$
Llama	$61.74_{.89}$	$.585_{.006}$	$52.34_{.72}$	$.511_{.007}$	$68.36_{.83}$	$.650_{.005}$	$71.62_{.76}$	$.711_{.007}$	$61.42_{.69}$	$.607_{.004}$	$\underline{74.52_{.71}}$	$\underline{.686_{.008}}$	$\mathbf{77.66_{.84}}$	$\mathbf{.742_{.003}}$
Mistral	$74.30_{.34}$	$.697_{.006}$	$71.52_{.56}$	$.666_{.003}$	$78.71_{.56}$	$.752_{.004}$	$84.29_{.56}$	$.811_{.005}$	$84.48_{.56}$	$.821_{.006}$	$\underline{85.23_{.56}}$	$\underline{.816_{.007}}$	$\mathbf{85.52_{.56}}$	$\mathbf{.828_{.004}}$
Avg.	68.10	.648	61.36	.587	72.38	.681	78.74	.762	75.78	.737	$\underline{80.83}$	$\underline{.766}$	**82.26**	**.790**
CARER														
Gemini	$59.20_{.51}$	$.493_{.004}$	$65.85_{.60}$	$.607_{.002}$	$\underline{70.85_{.52}}$	$\underline{.621_{.002}}$	$61.65_{.49}$	$.533_{.004}$	$59.95_{.67}$	$.541_{.004}$	$66.25_{.41}$	$.596_{.002}$	$\mathbf{72.65_{.67}}$	$\mathbf{.669_{.005}}$
Llama	$56.75_{.31}$	$.488_{.005}$	$\underline{65.70_{.60}}$	$\underline{.594_{.003}}$	$61.95_{.49}$	$.537_{.006}$	$57.35_{.32}$	$.499_{.002}$	$59.50_{.71}$	$.526_{.003}$	$64.25_{.54}$	$.579_{.004}$	$\mathbf{69.15_{.32}}$	$\mathbf{.635_{.002}}$
Mistral	$56.50_{.41}$	$.506_{.002}$	$67.10_{.48}$	$.617_{.004}$	$68.89_{.37}$	$.601_{.003}$	$60.25_{.29}$	$.515_{.003}$	$58.75_{.50}$	$.498_{.002}$	$\underline{72.10_{.43}}$	$\underline{.670_{.003}}$	$\mathbf{76.85_{.20}}$	$\mathbf{.717_{.004}}$
Avg.	57.48	.495	66.22	.606	67.23	.586	59.75	.516	59.40	.521	$\underline{67.53}$	$\underline{.615}$	**72.88**	**.674**
BBCNews														
Gemini	$87.00_{.31}$	$.869_{.013}$	$87.70_{.45}$	$.872_{.007}$	$\mathbf{90.99_{.21}}$	$\mathbf{.909_{.005}}$	$85.30_{.64}$	$.850_{.010}$	$86.20_{.35}$	$.858_{.011}$	$88.60_{.56}$	$.884_{.008}$	$\underline{89.50_{.37}}$	$\underline{.892_{.006}}$
Llama	$94.89_{.56}$	$.948_{.008}$	$93.43_{.41}$	$.933_{.004}$	$94.70_{.31}$	$.946_{.006}$	$93.60_{.41}$	$.935_{.004}$	$96.00_{.21}$	$.960_{.008}$	$\underline{96.10_{.52}}$	$\underline{.960_{.007}}$	$\mathbf{96.80_{.50}}$	$\mathbf{.967_{.005}}$
Mistral	$\underline{91.70_{.26}}$	$\underline{.915_{.005}}$	90.60_{31}	$.903_{.002}$	$\mathbf{92.99_{.29}}$	$\mathbf{.929_{.006}}$	$83.10_{.46}$	$.826_{.017}$	$83.00_{.41}$	$.825_{.007}$	$87.20_{.29}$	$.872_{.010}$	$88.10_{.49}$	$.879_{.009}$
Avg.	91.20	.910	90.57	.902	**92.89**	**.928**	87.33	.870	88.40	.881	90.63	.905	$\underline{91.47}$	$\underline{.912}$

Table 1: Text classification results evaluated on four datasets using three LLMs. **Bold** indicates the best result and underline indicates the result worse than the best result.

Method	Gemini		Llama		Mistral		Avg.	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
SST2								
Baseline	$94.67_{.47}$	$.947_{.002}$	$90.82_{.71}$	$.908_{.003}$	$94.34_{.29}$	$.943_{.003}$	93.27	.932
PLA	$\mathbf{95.44_{.28}}$	$\mathbf{.954_{.003}}$	$\underline{93.46_{.31}}$	$\underline{.934_{.004}}$	$94.34_{.36}$	$.943_{.005}$	94.41	.943
NLP	$\underline{95.38_{.30}}$	$\mathbf{.954_{.003}}$	$92.31_{.16}$	$.922_{.004}$	$\mathbf{96.37_{.46}}$	$\mathbf{.963_{.002}}$	$\underline{94.68}$	$\underline{.946}$
CTL	$\mathbf{95.44_{.35}}$	$\mathbf{.954_{.002}}$	$91.65_{.62}$	$.916_{.004}$	$95.11_{.28}$	$.951_{.003}$	94.06	.940
Ours	95.17_{37}	$\underline{.952_{.004}}$	$\mathbf{95.06_{.43}}$	$\mathbf{.951_{.004}}$	$\underline{95.60_{.21}}$	$\underline{.956_{.004}}$	**95.28**	**.953**
CoLA								
Baseline	$82.74_{.72}$	$.795_{.008}$	$64.52_{.71}$	$.586_{.008}$	$85.23_{.56}$	$.816_{.007}$	77.49	.732
PLA	$\underline{83.31}_{.54}$	$\underline{.798_{.006}}$	$\underline{73.53_{.86}}$	$\underline{.656_{.008}}$	$\underline{85.31_{.75}}$	$\mathbf{.832_{.008}}$	$\underline{80.72}$	$\underline{.762}$
NLP	$82.45_{.43}$	$.791_{.005}$	$64.05_{.79}$	$.579_{.008}$	$85.04_{.64}$	$.823_{.005}$	77.18	.731
CTL	$82.74_{.86}$	$.794_{.007}$	$62.79_{.62}$	$.579_{.004}$	$85.04_{.95}$	$.824_{.011}$	76.86	.732
Ours	$\mathbf{83.60_{.91}}$	$\mathbf{.801_{.006}}$	$\mathbf{77.66_{.84}}$	$\mathbf{.742_{.003}}$	$\mathbf{85.52_{.58}}$	$\underline{.828_{.004}}$	**82.26**	**.790**
CARER								
Baseline	$66.25_{.41}$	$.596_{.002}$	$64.25_{.54}$	$.579_{.004}$	$\underline{72.10_{.43}}$	$\underline{.670_{.003}}$	$\underline{67.53}$	$\underline{.615}$
PLA	$65.75_{.64}$	$.598_{.005}$	$61.65_{.52}$	$.556_{.011}$	$65.55_{.61}$	$.596_{.008}$	64.32	.583
NLP	$\underline{67.35_{.39}}$	$\underline{.619_{.004}}$	$64.40_{.25}$	$.583_{.007}$	$70.00_{.38}$	$.644_{.005}$	67.25	$\underline{.615}$
CTL	$66.90_{.45}$	$.605_{.007}$	$\underline{65.40_{.50}}$	$\underline{.586_{.005}}$	$67.80_{.44}$	$.615_{.007}$	66.70	.602
Ours	$\mathbf{72.65_{.67}}$	$\mathbf{.669_{.005}}$	$\mathbf{69.15_{.32}}$	$\mathbf{.635_{.002}}$	$\mathbf{76.85_{.20}}$	$\mathbf{.717_{.004}}$	**72.88**	**.673**
BBCNews								
Baseline	$88.60_{.56}$	$.884_{.008}$	$96.10_{.52}$	$.960_{.007}$	$87.20_{.29}$	$.872_{.010}$	90.63	.905
PLA	$89.40_{.35}$	$.891_{.005}$	$\underline{96.70_{.60}}$	$\underline{.966_{.003}}$	$\mathbf{89.50_{.29}}$	$\mathbf{.895_{.002}}$	$\underline{91.86}$	$\underline{.917}$
NLP	$89.00_{.37}$	$.889_{.002}$	$96.40_{.56}$	$.964_{.004}$	$88.40_{.42}$	$.883_{.006}$	91.20	.875
CTL	$\mathbf{90.30_{.54}}$	$\mathbf{.901_{.003}}$	$96.50_{.71}$	$.964_{.003}$	$\underline{89.40_{.63}}$	$\underline{.893_{.006}}$	**92.06**	**.919**
Ours	$\underline{89.50_{.37}}$	$\underline{.892_{.006}}$	$\mathbf{96.80_{.50}}$	$\mathbf{.967_{005}}$	$88.10_{.45}$	$.879_{.009}$	91.47	.912

Table 2: A comparative analysis of various linguistic label retrieval methods across four datasets.

5.1.3 Experimental Details

LLMs. We conduct experiments using three LLMs: Gemini (Team et al., 2024), Llama (Dubey et al., 2024) and Mistral (Jiang et al., 2024). Specifically, we utilize fixed versions of these models, namely Gemini 1.5 Flash, Llama 3.2-90b-Vision, and Mistral Large. These recently developed models demonstrate strong performance and exceptional generalization across a variety of tasks.

Similarity function. We define a similarity function, sim, as the cosine similarity between two sentence embeddings. These embeddings are generated using the all-MiniLM-L6-v2 model from the SBERT (Reimers and Gurevych, 2019).

Implementation details. For all LLMs, we use two random seeds and report the average results. We set the default number of demonstrations$k$ per class to 3 for all experiments. We adopt the typical prompt design methodology proposed by (Luo et al., 2024). To ensure accurate and consistent results in text classification tasks, we employ fixed hyperparameters for LLMs, thereby minimizing variability and limiting creative outputs. Further details are provided in Appendix B.

5.2 Experimental Results

5.2.1 Main results

Table 1 presents the results obtained using various retrieval strategies across three LLMs. The zero-shot approach, which does not rely on retrieving relevant demonstrations from the support set, leverages only the semantic understanding of labels. This strategy enables LLMs to achieve a baseline level of accuracy without additional context. Although Z-ICL mitigates the Copying Effect by leveraging physical neighbors and synonym labels, it only marginally outperforms the zero-shot baseline. However, it lags behind other methods, likely due to the inherent complexity and challenges associated with selecting appropriate synonym labels. KATE achieves better performance than zero-shot and Z-ICL by utilizing the most similar demonstrations to the query. However, it is susceptible to errors caused by misleading similarities. As a result, KATE still struggles to perform well on the CoLA and CARER datasets. To mitigate the effects of misleading similarities, NwayKshot generally outperforms KATE in most scenarios. However, as

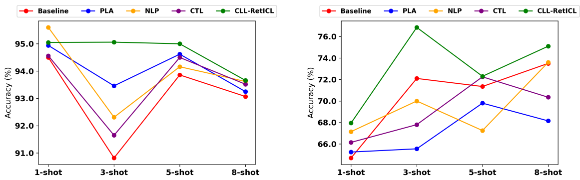


Figure 4: A comparison of the performance of various shot configurations is presented across a baseline and four linguistic label retrieval strategies. Evaluations for the SST2 task (using Llama) are on the left, while results for the CARER task (using Mistral) appear on the right.

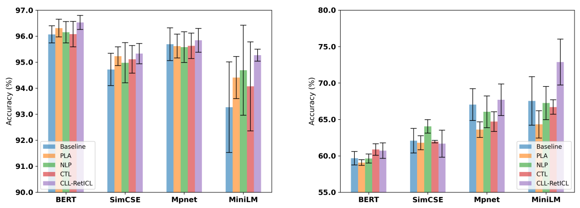


Figure 5: A comparison of the performance of various sentence embedding models is presented, with evaluations conducted on SST2 on the left and CARER on the right.

noted earlier, NwayKshot still struggles to identify an optimal combination of demonstrations. VoteK attempts to further select more effective and relevant demonstrations from the support set. However, this method still fails to utilize label information effectively. On the other hand, Cluster-TopN leverages label information from a distributional perspective but does not account for the linguistic meaning of the labels. While both VoteK and Cluster-TopN show improvements in accuracy for certain tasks, they fall short in addressing a fundamental issue: the importance of linguistic label meaning in text classification tasks. This oversight leads to inconsistent performance and highlights their inherent weaknesses. Finally, our proposed method, CLL-RetICL, significantly outperforms all baseline approaches. On average, CLL-RetICL improves RetICL's performance by an absolute margin of 2–15% over the zero-shot strategy and by 0.57–13.48% over existing RetICL-based methods. These results demonstrate consistent performance gains across all datasets and LLMs by effectively leveraging the relationships between linguistic labels and their corresponding sentences.

Comparison to Variants of Label-Related RetICL. We use the NwayKshot method as our baseline, a retrieval-based approach that does not utilize linguistic label information. To enhance performance, we evaluate four proposed strategies that incorporate linguistic label related retrieval methods, with the results summarized in Table 2. All four strategies outperform the baseline across all datasets and LLMs, demonstrating the benefits of leveraging label information. Among these, CLL-RetICL consistently delivers the best performance, achieving an average absolute improvement of 0.8–5.3% over the NwayKshot method. While PLA, NLP, and CTL also surpass the baseline, they show minor performance drops on certain tasks. In contrast, CLL-RetICL not only outperforms these methods in most tasks but also achieves consistent gains in classification accuracy.

5.3 Ablation Study

We conduct detailed ablation studies to analyze the significance of each component in CLL-RetICL. In our ablation study, the NwayKshot approach serves as the baseline, as shown in the following tables and figures.

Effect of the number of shots. The number of shots significantly impacts the performance of LLMs. We explore experiments comparing four different shot configurations for each label class: 1-shot, 3-shot, 5-shot, and 8-shot. Figure 4 presents partial results, while the complete results are provided in Appendix D.1. The results in Figure 4 demonstrate that CLL-RetICL consistently outperforms the baseline methods across different values of k. While some alternative strategies occasionally achieve better performance than CLL-RetICL, they lack robustness and often fall short of both CLL-RetICL and the baselines. This indicates that CLL-RetICL delivers more stable performance across a range of scenarios. Based on the experimental results, we selected $k = 3$ as the hyperparameter for the number of shots, as CLL-RetICL demonstrated higher improvement with a 3-shot configuration.

Effect of sentence embedding model. Pre-trained sentence embeddings play a crucial role in ICL. The objective is to evaluate the effectiveness of the proposed methods by comparing them against four off-the-shelf sentence embedding models. Figure 5 illustrates the average performance of three LLMs across two datasets. CLL-RetICL consistently outperforms the baseline and the other three strategies across all sentence embedding models, with the exception of SimCSE (Gao et al., 2021) in the CARER dataset. We attribute the relatively lower performance of our method with

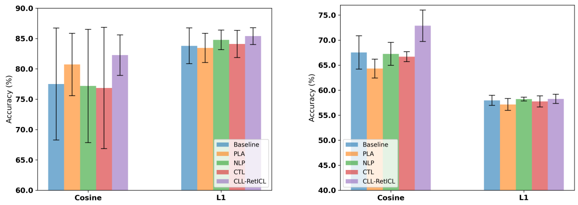


Figure 6: A comparison of the performance of various similarity functions is presented, with evaluations conducted on CoLA on the left and CARER on the right.

SimCSE to the fact that SimCSE has already employed contrastive learning to fine-tune the pre-trained sentence embedding model. This suggests that our approach is generally more effective for pre-trained sentence embeddings that do not utilize contrastive learning strategies. Compared to other sentence embedding models, MiniLM demonstrates the greatest improvement over the baseline; therefore, we have chosen it as our default. Full results are presented in Appendix D.2.

Effect of similarity function. To evaluate the effect of the similarity function in our CLL-RetICL model, we compare its performance using another similarity function, L1, as described in (Winata et al., 2023). The results are presented in Figure 6 with detailed results provided in Appendix D.3.

CLL-RetICL performs effectively with both cosine and L1 similarity functions. However, experiments show that cosine similarity outperforms the L1 function, suggesting that it better leverages CLL-RetICL's potential. Consequently, we use cosine similarity as the default.

Effect of w/o similarity between demonstration and query. Because our proposed additional component can serve as a scoring criterion for selecting demonstrations, the question arises whether the similarity score between demonstrations and the query should be included in CLL-RetICL.

We evaluate the problem and present the results in Figure 7. Our findings indicate that the performance without the component addressing the similarity between queries and sentences is consistently lower than that of linguistically labeled RetICL. In fact, it performs even worse than the baseline. These results highlight that the similarity component between queries and sentences is an essential part of the retrieval process. Detailed results are presented in Appendix D.4.

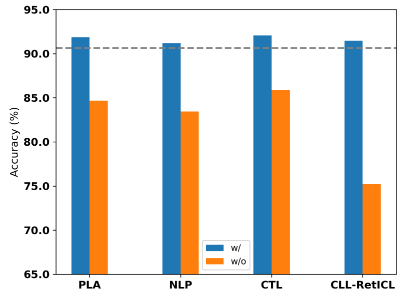


Figure 7: A comparison of the retrieval process with and without incorporating the similarity score between the query and the sentence is illustrated on BBCNews dataset. The baseline is represented by a dashed line.

Effect of trade-off hyperparameters. We use a trade-off approach to balance the impact between sentences and their label set. Based on the results of the previous experiment, sentence-query similarity remains a crucial factor in selecting relevant demonstrations. This raises an important question: how should we trade off between the original method, which retrieves the closest demonstrations to the query, and our approach? To address this question, we evaluate the effects of various hyperparameter settings. Specifically, we focus on hyperparameters lower than 1.0, as previous research has consistently shown that closer demonstrations generally outperform those that are further away. We maintain the principle that proximity to the query remains a core factor in our approach. Based on these observations in Appendix D.5, we found that the trade-off hyperparameter has some influence on the final results. However, their impact on PLA, NLP, and CTL methods is relatively small. Interestingly, we observed that a trade-off hyperparameter value of 1.0 yields the best performance for our CLL-RetICL method. Consequently, we adopt 1.0 as the default hyperparameter.

6 Conclusion

This paper introduces a new paradigm Contrastive Linguistic Label Retrieval-based In-Context Learning. Unlike existing approaches that universally sample demonstrations without considering the linguistic label information, we propose a general framework for identifying more effective and relevant demonstrations. This framework enhances the capabilities of LLMs to produce more accurate text classification results. Additionally, we design a universal prompt that is adaptable to all text classification tasks. Empirical evaluation on four datasets demonstrates that CLL-RetICL significantly outperforms conventional practices in RetICL by incorporating the similarity between linguistic labels and sentences. This highlights the promising performance of CLL-RetICL and opens up several intriguing research opportunities for further methodological exploration.

7 Limitations

Requiring Semantic Labels. Our approach focuses exclusively on the semantic label text classification task. Certain text classification scenarios, however, may involve ambiguous label classes, such as class0, class1, ... Ambiguities in labeling may introduce additional challenges and Addressing these issues remains an open area for future research.

Handling complex classification tasks with ambiguous labels presents additional challenges for our method, as CLL-RetICL relies heavily on semantic label representations. To illustrate this issue, we use the TREC dataset (Li and Roth, 2002; Hovy et al., 2001), which provides both abbreviated and full-form class labels. In our analysis, we adopt the coarse-label scheme and specifically compare the abbreviated class labels with their corresponding full descriptive labels. The abbreviated labels include [ABBR, ENTY, DESC, HUM, LOC, NUM], while their full counterparts are [Abbreviation, Entity, Description and abstract concept, Human being, Location, Numeric value]. The results, shown in Table 3, demonstrate that using abbreviated labels weakens the performance of our method compared to using the full descriptive labels.

TREC	Abbr.		Full	
	ACC	F1	ACC	F1
	Llama			
Nwaykshot	**57.40**	**0.591**	56.60	0.577
CLL-RetICL	56.40	0.603	**57.60**	**0.603**

Table 3: A comparison of classification accuracy (%) and F1 score to evaluate the impact of abbreviated labels versus full labels on the TREC dataset.

Better Descriptive Labels Recently, the use of class-label synonyms has become a popular and compelling topic of research (Pawar et al., 2024). In our work, we also present results using class-label synonyms on the SST-2 dataset, as shown in Table 4. Our findings indicate that CLL-RetICL consistently performs well across different label synonym settings. However, the overall performance with label synonym settings is lower compared to using the original labels. These results suggest that more accurate, semantic, and suitable labels could further enhance the effectiveness of our method.

Moreover, some classification tasks include explanations for the meaning of each label. Using more descriptive sentences and designing multi-label descriptors can help reduce the risk of bias and support effective mitigation strategies. In this work, we did not utilize those explanations. Incorporating these explanations into the classification process is left as a direction for future work.

SST2	Positive/Negative		Great/Terrible		Good/Bad	
	ACC	F1	ACC	F1	ACC	F1
	Llama					
Nwaykshot	90.82	0.908	92.48	0.927	91.98	0.923
CLL-RetICL	**95.06**	**0.951**	**93.90**	**0.939**	**94.23**	**0.944**

Table 4: A comparison of classification accuracy (%) and F1 score to evaluate the impact of synonym labels on the SST2 dataset. The synonym pairs used in this study are drawn from previously published work (Pawar et al., 2024).

Enhance prompt clarity. In previous work, researchers observed that well-crafted prompts can lead to better results. However, in this study, we did not compare the effects of different prompt formats. Determining how to construct optimal prompts to leverage the potential of our CLL-RetICL framework fully remains an open question and is left for future exploration.

References

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.

Youngjin Chae and Thomas Davidson. 2023. Large language models for text classification: From zero-shot learning to fine-tuning. *Open Science Foundation*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Aleksandra Edwards and Jose Camacho-Collados. 2024. Language models for text classification: Is in-context learning enough? *arXiv preprint arXiv:2403.17661*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

John Fields, Kevin Chovanec, and Praveen Madiraju. 2024. A survey of text classification with transformers: How wide? how large? how long? how accurate? how expensive? how safe? *IEEE Access*.

Lingyu Gao, Aditi Chaudhary, Krishna Srinivasan, Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. 2023. Ambiguity-aware in-context learning with large language models. *arXiv preprint arXiv:2309.07900*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. 2024. What makes a good order of examples in in-context learning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14892–14904.

Sabit Hassan and Malihe Alikhani. 2023. D-calm: A dynamic clustering-based active learning approach for mitigating bias. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5540–5553.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Hongpeng Jin, Wenqi Wei, Xuyu Wang, Wenbin Zhang, and Yanzhao Wu. 2023. Rethinking learning rate tuning in the era of large language models. In *2023 IEEE 5th International Conference on Cognitive Machine Intelligence (CogMI)*, pages 112–121. IEEE.

Guozheng Li, Peng Wang, Jiajun Liu, Yikai Guo, Ke Ji, Ziyu Shang, and Zijie Xu. 2024. Meta in-context learning makes large language models better zero and few-shot relation extractors. *arXiv preprint arXiv:2404.17807*.

Jing Li, Billy Chiu, Shanshan Feng, and Hao Wang. 2020. Few-shot named entity recognition via meta-learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(9):4245–4256.

Junlong Li, Jinyuan Wang, Zhuosheng Zhang, and Hai Zhao. 2022. Self-prompting large language models for zero-shot open-domain qa. *arXiv preprint arXiv:2212.08635*.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. *arXiv preprint arXiv:2305.04320*.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. *arXiv preprint arXiv:2302.13539*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

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 (Volume 1: Long Papers)*, pages 8086–8098.

Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. 2024. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*.

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*.

Costas Mavromatis, Balasubramaniam Srinivasan, Zhengyuan Shen, Jiani Zhang, Huzefa Rangwala, Christos Faloutsos, and George Karypis. 2023. Which examples to annotate for in-context learning? towards effective and efficient selection. *arXiv preprint arXiv:2310.20046*.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. 2023. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 173–184.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Sachin Pawar, Nitin Ramrakhiyani, Anubhav Sinha, Manoj Apte, and Girish Palshikar. 2024. Why generate when you can discriminate? a novel technique for text classification using language models. In *Findings of the association for computational linguistics: EACL 2024*, pages 1099–1114.

Branislav Pecher, Ivan Srba, Maria Bielikova, and Joaquin Vanschoren. 2024. Automatic combination of sample selection strategies for few-shot learning. *arXiv preprint arXiv:2402.03038*.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Hamidreza Rouzegar and Masoud Makrehchi. 2024. Enhancing text classification through llm-driven active learning and human annotation. *arXiv preprint arXiv:2406.12114*.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3687–3697.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

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.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts. *arXiv preprint arXiv:2209.11055*.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. *arXiv preprint arXiv:2305.14160*.

Zhiqiang Wang, Yiran Pang, and Yanbin Lin. 2024. Smart expert system: Large language models as text classifiers. *arXiv preprint arXiv:2405.10523*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Genta Indra Winata, Liang-Kang Huang, Soumya Vadlamannati, and Yash Chandarana. 2023. Multilingual few-shot learning via language model retrieval. *arXiv preprint arXiv:2306.10964*.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2022. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. *arXiv preprint arXiv:2212.10375*.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1423–1436.

Zhichao Xu, Daniel Cohen, Bei Wang, and Vivek Srikumar. 2024. In-context example ordering guided by label distributions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2623–2640.

Hao Yu, Zachary Yang, Kellin Pelrine, Jean Francois Godbout, and Reihaneh Rabbany. 2023. Open, closed, or small language models for text classification? *arXiv preprint arXiv:2308.10092*.

Yazhou Zhang, Mengyao Wang, Chenyu Ren, Qiuchi Li, Prayag Tiwari, Benyou Wang, and Jing Qin. 2024. Pushing the limit of llm capacity for text classification. *arXiv preprint arXiv:2402.07470*.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022a. Active example selection for in-context learning. *arXiv preprint arXiv:2211.04486*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

Fedor Zhdanov. 2019. Diverse mini-batch active learning. *arXiv preprint arXiv:1901.05954*.

A Data Statistics

We take the four text classification tasks including SST2, CoLA, CARER, and BBCNews. See the descriptions and statistics in Table 5.

We use the original SST-2 dataset that only comprises the complete sentences that are not labeled neutral, and its original split is 6920/872/1821 (Socher et al., 2013).

CoLA comprises 10,657 sentences sourced from 23 linguistics publications. Each sentence has been expertly annotated for acceptability (i.e., grammaticality) by the original authors (Warstadt et al., 2018). CoLA is divided into two subsets: a training set and a development set. In our work, we treat the development set as the test set.

CARER is a dataset of English Twitter messages with six basic emotions: anger, fear, joy, love, sadness, and surprise (Saravia et al., 2018). The original CARER dataset has been split into trainsets, validation, and test sets.

The BBC News Topic Classification dataset consists of 2,225 articles published on the BBC News website between 2004 and 2005. Each article is labeled under one of 5 categories: business, entertainment, politics, sport, or tech (Greene and Cunningham, 2006). The original BBCNews dataset has been split into trainsets and test sets.

As stated in the main text, we exclude the validation set.

B Implementation Details

All implementations are done in PyTorch.

Prompt template. We adopt the prompt used in the CARER task as a template. Following the approach of Luo et al. (2024), we design our prompt for universal text classification tasks, as shown in Table 7. $(demo\_1)$, $(demo\_2)$, $(demo\_3)$ are selected demonstration from support set. $(query)$ is the current query sentence.

Budget. We conducted experiments on LLMs across four public datasets, utilizing APIs to compute the results. To ensure consistency and avoid generating creative outputs, we fixed the LLMs' hyperparameters, as detailed in Table 6. The total cost of running these experiments through the APIs amounted to approximately 1,000 US dollars.

C Linguistic Label Retrieval Hypothesis

To explore the question: Are the clustering centers reliable, we analyze the distribution of clustering centers, as shown in Figure 8.

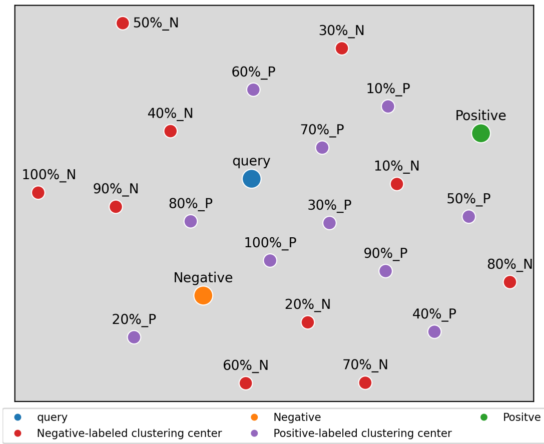


Figure 8: An example illustrating the distribution of queries, linguistic labels, and clustering centers in a pre-trained sentence embedding model using t-SNE. $10\%\_N$ and $10\%\_P$ represent a pair, indicating that 10% of the support set is used to estimate the clustering center. In this notation, "N" refers to the Negative-labeled clustering center, while "P" denotes the Positive-labeled clustering center.

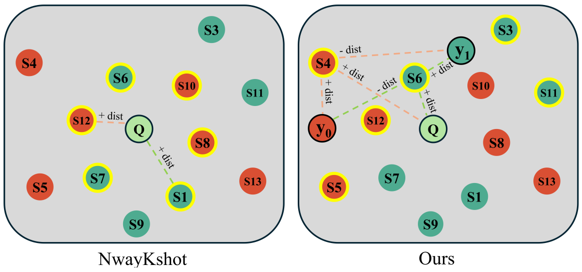


Figure 9: A comparison of the retrieved demonstrations between NwayKshot and our method in the sentence embedding vector space. A yellow circle indicates a selected sentence.

We present five examples in Table 8 to illustrate our findings. The experiment was conducted on the SST2 dataset, where we treated the training set as the support set and the test set as the query set. For each query example in the test set, we provide its index and the indices of the selected demonstrations from the training set. Additionally, we report the similarity scores calculated using cosine distance within the vector space of a pre-trained sentence embedding model (Reimers and Gurevych, 2019).

D Additional Results

D.1 Effect of the number of shots.

We present the detailed results in Table 9.

Dataset	Trainset	Testset	Label
SST2	6,920	1,821	"Negative","Positive"
CoLA	8,551	1,043	"Unacceptable", "Acceptable"
CARER	16,000	2,000	"Sadness", "Joy", "Love", "Anger", "Fear", "Surprise"
BBCNews	1,225	1,000	"Business", "Entertainment", "Politics", "Sport", "Tech"

Table 5: Statistics of datasets as well as their labels.

Configure	Gemini	Llama	Mistral
$"temperature"$	0.2	0.01	0.01
$"top\_p"$	0.9	0.9	0.5
$"top\_k"$	1	1	1
$"max\_output\_tokens"$	2	2	2

Table 6: generation_configure of hyperparameters in various LLMs.

D.2 Effect of sentence embedding model.

We present the detailed results in Table 10.

D.3 Effect of similarity function.

We present the detailed results in Table 11.

D.4 Effect of w/o similarity between demonstration and query.

We present the detailed results in Table 14.

D.5 Effect of trade-off hyperparameters.

We present the detailed results in Table 12, Table 13, Table 15, Table 16, Table 17.

System message	"You are given a task where there are multiple classes, and for each class, a few labeled examples are provided. Based on these examples, you need to classify a new unseen instance. Choose ONLY one tag and output the tag. Do Not output others."
	CARER
Prompt	Class0: sadness 1. Example 1: (demo_1) -> "sadness" 2. Example 2: (demo_2) -> "sadness" 3. Example 3: (demo_3) -> "sadness" Class1: joy 1. Example 1: (demo_1) -> "joy" 2. Example 2: (demo_2) -> "joy" 3. Example 3: (demo_3) -> "joy" Class2: love 1. Example 1: (demo_1) -> "love" 2. Example 2: (demo_2) -> "love" 3. Example 3: (demo_3) -> "love" Class3: anger 1. Example 1: (demo_1) -> "anger" 2. Example 2: (demo_2) -> "anger" 3. Example 3: (demo_3) -> "anger" Class4: fear 1. Example 1: (demo_1) -> "fear" 2. Example 2: (demo_2) -> "fear" 3. Example 3: (demo_3) -> "fear" Class5: surprise 1. Example 1: (demo_1) -> "surprise" 2. Example 2: (demo_2) -> "surprise" 3. Example 3: (demo_3) -> "surprise" Query: (query) Prediction:

Table 7: Designed a universal prompt for all text classification tasks.

query	label	Not Correct						Correct					
		index	sentence	label_N	label_P	center_N	center_P	index	sentence	label_N	label_P	center_N	center_P
36	N	3465	0.501	0.111	**0.187**	0.576	**0.626**	1344	0.379	**0.173**	0.087	**0.432**	0.417
		5169	0.447	0.095	**0.162**	0.434	**0.454**	5012	0.399	**0.124**	0.063	**0.656**	0.613
		4441	0.436	0.096	**0.131**	0.477	**0.526**	6432	0.399	**0.188**	0.131	**0.461**	0.450
	P	5529	0.603	**0.232**	0.187	0.393	**0.479**	4310	0.580	0.170	**0.247**	0.363	**0.444**
		4310	0.580	**0.247**	0.170	0.363	**0.444**	5529	0.603	0.187	**0.231**	0.393	**0.479**
		5879	0.507	**0.175**	0.106	0.561	**0.646**	6723	0.427	0.129	**0.291**	0.442	**0.591**
49	N	4084	0.694	**0.022**	−0.033	**0.525**	0.458	4084	0.694	**0.022**	−0.033	**0.525**	0.458
		3465	0.564	0.111	**0.187**	0.576	**0.626**	6331	0.562	**0.122**	0.015	**0.647**	0.569
		6331	0.562	**0.122**	0.015	**0.647**	0.569	4290	0.543	**0.093**	0.044	**0.583**	0.569
	P	1625	0.672	0.072	**0.134**	0.531	**0.567**	1625	0.672	0.072	**0.134**	0.531	**0.567**
		4273	0.576	**0.040**	0.038	0.533	**0.579**	1268	0.506	−0.024	**0.136**	0.428	**0.482**
		1936	0.565	**0.669**	0.103	0.510	**0.550**	543	0.516	0.126	**0.272**	0.379	**0.448**
1690	N	1613	0.569	**-0.029**	−0.053	**0.341**	0.321	1613	0.569	**-0.029**	−0.053	**0.341**	0.321
		5550	0.520	**0.025**	0.015	**0.283**	0.247	4127	0.497	**0.037**	−0.033	**0.271**	0.250
		4127	0.497	**0.037**	−0.033	**0.271**	0.250	801	0.466	**0.127**	0.047	**0.464**	0.396
	P	3043	0.600	0.019	**0.056**	0.194	**0.217**	3043	0.600	0.019	**0.056**	0.194	**0.217**
		444	0.502	**0.112**	0.093	0.334	**0.338**	4941	0.401	−0.029	**0.091**	0.464	**0.562**
		1856	0.480	−0.008	**-0.004**	0.327	**0.363**	1856	0.480	−0.008	**-0.004**	0.327	**0.363**
1694	N	2433	0.545	−0.054	**0.038**	0.434	**0.441**	3367	0.485	**0.092**	0.061	**0.586**	0.540
		3367	0.485	**0.092**	0.061	**0.586**	0.540	4925	0.478	**-0.002**	−0.041	**0.507**	0.456
		4925	0.478	**-0.002**	−0.041	**0.507**	0.456	3643	0.428	**0.057**	−0.026	**0.557**	0.543
	P	613	0.455	−0.031	**-0.011**	0.356	**0.405**	6713	0.433	0.110	**0.258**	0.549	**0.615**
		324	0.447	**0.181**	0.105	0.572	**0.623**	5337	0.424	0.056	**0.188**	0.337	**0.478**
		5135	0.446	0.114	**0.198**	0.470	**0.572**	5135	0.446	0.114	**0.198**	0.470	**0.572**
1809	N	5557	0.512	**0.168**	0.138	**0.321**	0.292	5557	0.512	**0.168**	0.138	**0.321**	0.292
		2756	0.405	0.045	**0.061**	**0.328**	0.336	4071	0.388	**0.114**	0.099	**0.402**	0.376
		2690	0.397	0.088	**0.910**	**0.566**	0.507	1430	0.382	**0.038**	0.023	0.331	**0.332**
	P	2193	0.480	0.090	**0.130**	**0.570**	0.567	2193	0.480	0.090	**0.130**	**0.570**	0.567
		6385	0.465	**0.094**	0.046	0.470	**0.477**	296	0.347	0.018	**0.135**	0.295	**0.417**
		679	0.391	**0.149**	0.095	0.427	**0.444**	897	0.359	0.162	**0.254**	0.394	**0.398**

Table 8: Five examples comparing incorrect demonstration combinations with their correct counterparts, as evaluated on SST2 task. In this notation, "N" refers to the "Negative" label, while "P" denotes the "Positive" label.

Method	1-shot			3-shot			5-shot			8-shot		
	Gemini	Llama	Mistral	Gemini	Llama	Mistral	Gemini	Llama	Mistral	Gemini	Llama	Mistral
SST2												
Baseline	94.67	94.50	94.61	94.67	90.82	94.34	95.16	93.86	95.00	95.60	93.07	95.00
PLA	94.78	94.94	94.34	**95.44**	93.46	94.34	95.00	94.62	95.22	95.33	93.35	95.00
NLP	**95.38**	**95.60**	**96.59**	95.38	92.31	**96.37**	**95.94**	94.16	95.44	95.00	93.62	95.10
CTL	94.28	94.56	94.28	**95.44**	91.65	95.11	95.28	94.50	94.89	95.00	93.52	**95.16**
CLL-RetICL	94.89	95.05	95.33	95.17	**95.06**	95.60	95.28	**95.00**	**95.71**	**96.21**	**93.66**	**95.16**
CARER												
Baseline	66.55	63.45	64.70	66.25	64.25	72.10	68.23	70.95	71.35	69.50	69.35	73.50
PLA	63.80	60.30	65.25	65.75	61.65	65.55	67.70	62.10	69.80	67.10	65.40	68.15
NLP	66.30	64.60	67.15	67.35	64.40	70.00	70.31	65.35	67.25	69.69	**72.30**	73.60
CTL	64.30	61.20	66.15	66.90	65.40	67.80	68.61	68.05	72.25	68.30	68.65	70.35
CLL-RetICL	**66.75**	**65.50**	**67.95**	**72.65**	**69.15**	**76.85**	**69.05**	**74.45**	**72.30**	**70.75**	71.35	**75.10**

Table 9: Full results of various shots effect in our proposed methods.

Approach	Bert				Simcse				Mpnet				MiniLM			
	Gemini	Llama	Mistral	Avg.	Gemini	Llama	Mistral	Avg.	Gemini	Llama	Mistral	Avg.	Gemini	Llama	Mistral	Avg.
SST2																
Baseline	95.93	95.76	96.54	$96.07_{0.33}$	95.11	93.85	95.21	$94.72_{0.62}$	95.66	94.93	**96.48**	$95.69_{0.63}$	94.67	90.82	94.34	$93.27_{1.74}$
PLA	95.93	96.26	96.76	$96.31_{0.34}$	94.78	**95.27**	95.66	$95.23_{0.36}$	**95.71**	95.02	96.15	$95.62_{0.46}$	**95.44**	93.46	94.34	$94.41_{0.81}$
NLP	96.26	95.60	96.59	$96.15_{0.41}$	95.38	93.90	95.66	$94.98_{0.77}$	95.55	94.87	96.32	$95.58_{0.59}$	95.38	92.31	**96.37**	$94.68_{1.73}$
CTL	95.88	95.60	96.76	$96.08_{0.49}$	94.94	94.56	**95.82**	$95.11_{0.53}$	95.40	95.18	96.32	$95.63_{0.49}$	**95.44**	91.65	95.11	$94.06_{1.71}$
CLL-RetICL	**96.32**	**96.37**	**96.92**	$\mathbf{96.53}_{0.27}$	**95.77**	94.83	95.39	$\mathbf{95.33}_{0.39}$	95.44	**95.60**	**96.48**	$\mathbf{95.84}_{0.46}$	95.17	**95.06**	95.60	$\mathbf{95.28}_{0.23}$
CARER																
Baseline	58.65	59.45	60.90	$59.67_{0.93}$	**64.45**	60.60	61.20	$62.08_{1.69}$	63.95	68.30	68.85	$67.03_{2.19}$	66.25	64.25	72.10	$67.53_{3.33}$
PLA	58.75	58.85	59.65	$59.08_{0.40}$	62.65	62.30	60.45	$61.80_{0.96}$	62.50	63.25	65.05	$63.60_{1.07}$	65.75	61.65	65.55	$64.32_{1.88}$
NLP	59.10	59.30	60.50	$59.63_{0.62}$	63.10	**63.75**	**65.30**	$\mathbf{64.05}_{0.92}$	63.00	67.15	68.00	$66.05_{2.18}$	67.35	64.40	70.00	$67.25_{2.29}$
CTL	59.80	**61.12**	61.70	$\mathbf{60.87}_{0.79}$	61.90	61.75	62.20	$61.95_{0.18}$	62.90	65.05	66.15	$64.70_{1.35}$	66.90	65.40	67.80	$66.70_{0.99}$
CLL-RetICL	**59.90**	60.05	**62.20**	$60.72_{1.05}$	59.05	62.80	63.15	$61.67_{1.86}$	**64.65**	**69.15**	**69.30**	$\mathbf{67.70}_{2.16}$	**72.65**	**69.15**	**76.85**	$\mathbf{72.88}_{3.14}$

Table 10: A Comparison of various pre-trained sentence embedding models.

Approach	Cosine				L1			
	Gemini	Llama	Mistral	Avg.	Gemini	Llama	Mistral	Avg.
				CoLA				
Baseline	82.74	64.52	85.23	77.50$_{9.23}$	85.43	79.65	86.28	83.79$_{2.95}$
PLA	83.31	73.53	85.31	80.72$_{5.14}$	84.66	80.09	85.62	83.46$_{2.41}$
NLP	82.45	64.05	85.04	77.18$_{9.34}$	85.13	82.64	86.57	84.78$_{1.62}$
CTL	82.74	62.79	85.04	76.86$_{9.99}$	84.56	81.17	86.57	84.10$_{2.23}$
CLL-RetICL	**83.60**	**77.66**	**85.52**	**82.26**$_{3.34}$	**85.81**	**83.51**	**86.86**	**85.39**$_{1.39}$
				CARER				
Baseline	66.25	64.25	72.10	67.53$_{3.33}$	57.30	57.20	59.40	57.97$_{1.01}$
PLA	65.75	61.65	65.55	64.32$_{1.88}$	55.95	56.75	58.75	57.15$_{1.18}$
NLP	67.35	64.40	70.00	67.25$_{2.29}$	**57.95**	**58.75**	57.95	58.22$_{0.38}$
CTL	66.90	65.40	67.80	66.70$_{0.99}$	56.75	57.20	59.30	57.75$_{1.11}$
CLL-RetICL	**72.65**	**69.15**	**76.85**	**72.88**$_{3.14}$	**57.95**	57.30	**59.50**	**58.25**$_{0.92}$

Table 11: A Comparison of Similar Function Methods

CTL	ACC								
LLM	(0.3, 0.3)	(0.3, 0.5)	(0.3, 1.0)	(0.5, 0.3)	(0.5, 0.5)	(0.5, 1.0)	(1.0, 0.3)	(1.0, 0.5)	(1.0, 1.0)
					CoLA				
Gemini	82.92	83.30	**84.35**	83.39	83.40	84.05	83.84	83.76	83.60
Llama	62.24	63.44	62.73	63.26	63.44	63.53	65.45	**64.04**	62.79
Mistral	**85.91**	85.33	85.90	85.71	85.62	85.33	85.71	85.33	85.04
					CARER				
Gemini	65.55	65.80	65.45	**67.00**	66.50	65.35	63.65	63.65	66.90
Llama	64.35	62.95	63.30	**68.25**	67.75	63.15	63.10	63.05	65.40
Mistral	68.30	67.75	67.10	**71.10**	70.90	66.60	65.10	64.95	67.80

Table 12: A comparison of classification accuracy (%) to assess the impact of various trade-off hyperparameters in the CTL strategy.

CTL	F1								
LLM	(0.3, 0.3)	(0.3, 0.5)	(0.3, 1.0)	(0.5, 0.3)	(0.5, 0.5)	(0.5, 1.0)	(1.0, 0.3)	(1.0, 0.5)	(1.0, 1.0)
					CoLA				
Gemini	0.791	0.795	**0.809**	0.798	0.797	0.806	0.802	0.801	0.801
Llama	0.552	0.584	0.570	0.581	0.581	0.589	**0.603**	0.588	0.579
Mistral	**0.825**	0.817	0.824	0.823	0.821	0.819	0.820	0.816	0.824
					CARER				
Gemini	0.594	0.596	0.595	**0.612**	0.607	0.590	0.575	0.573	0.605
Llama	0.576	0.559	0.567	**0.612**	0.602	0.570	0.570	0.566	0.586
Mistral	0.631	0.605	0.607	**0.648**	0.645	0.595	0.585	0.582	0.615

Table 13: A comparison of F1 score (%) to assess the impact of various trade-off hyperparameters in the CTL strategy.

	Gemini		Llama		Mistral		Avg.	
Method	ACC	F1	ACC	F1	ACC	F1	ACC	F1
				CARER				
Baseline	66.25	0.596	64.25	0.579	72.10	0.670	67.53	0.615
PLA	**65.75**	**0.598**	**61.65**	**0.556**	**65.55**	**0.596**	**64.32**	**0.583**
w/o	57.80	0.514	56.75	0.497	56.10	0.488	56.88	0.499
NLP	**67.35**	**0.619**	**64.40**	**0.583**	**70.00**	**0.644**	**67.25**	**0.615**
w/o	57.05	0.502	58.90	0.512	59.70	0.529	58.55	0.514
CTL	**66.90**	**0.605**	**65.40**	**0.586**	**67.80**	**0.615**	**66.70**	**0.602**
w/o	58.40	0.521	56.30	0.494	57.75	0.505	57.48	0.507
Ours	**72.65**	**0.669**	**69.15**	**0.635**	**76.85**	**0.717**	**72.88**	**0.673**
w/o	52.25	0.468	55.50	0.486	51.70	0.463	53.15	0.472
				BBCNews				
Baseline	88.60	0.884	96.10	0.960	87.20	0.872	90.63	0.905
PLA	**89.40**	**0.891**	**96.70**	**0.966**	**89.50**	**0.895**	**91.86**	**0.917**
w/o	79.50	0.777	94.20	0.940	80.30	0.796	84.67	0.837
NLP	**89.00**	**0.889**	**96.40**	**0.964**	**88.40**	**0.883**	**91.20**	**0.875**
w/o	84.60	0.843	80.20	0.801	85.50	0.854	83.43	0.832
CTL	**90.30**	**0.901**	**96.50**	**0.964**	**89.40**	**0.893**	**92.06**	**0.919**
w/o	83.40	0.822	94.20	0.942	80.10	0.792	85.90	0.852
Ours	**89.50**	**0.892**	**96.80**	**0.967**	**88.10**	**0.879**	**91.47**	**0.912**
w/o	77.00	0.750	70.10	0.698	78.50	0.770	75.20	0.739

Table 14: A comparison of the retrieval process with and without incorporating the similarity score between the query and sentence.

CLL-RetICL	0.3		0.5		0.7		1.0	
LLM	ACC	F1	ACC	F1	ACC	F1	ACC	F1
				CoLA				
Gemini	82.92	0.791	83.21	0.794	83.01	0.793	**83.60**	**0.801**
Llama	77.60	0.746	**77.68**	**0.757**	76.53	0.737	77.66	0.742
Mistral	85.43	0.818	85.33	0.817	**85.71**	0.822	85.52	**0.828**
				CARER				
Gemini	69.10	0.636	69.85	0.640	70.10	0.640	**72.65**	**0.669**
Llama	68.50	0.625	68.65	0.625	**69.95**	**0.635**	69.15	**0.635**
Mistral	72.20	0.665	72.20	0.671	71.65	0.656	**76.85**	**0.717**

Table 15: A comparison of classification accuracy (%) and F1 score to assess the impact of various trade-off hyperparameters in CLL-RetICL strategy.

PLA	0.3		0.5		0.7		1.0	
LLM	ACC	F1	ACC	F1	ACC	F1	ACC	F1
				CoLA				
Gemini	**83.57**	**0.799**	82.42	0.784	83.17	0.794	83.31	0.798
Llama	73.12	0.661	73.03	0.649	**74.29**	**0.681**	73.53	0.656
Mistral	**85.71**	0.821	85.42	0.817	85.23	0.813	85.31	**0.832**
				CARER				
Gemini	66.60	0.602	**66.70**	**0.603**	65.85	0.598	65.75	0.598
Llama	**65.15**	**0.585**	64.90	**0.585**	62.50	0.562	61.65	0.556
Mistral	**65.80**	0.595	65.15	0.579	62.50	0.558	65.55	**0.596**

Table 16: A comparison of classification accuracy (%) and F1 score to assess the impact of various trade-off hyperparameters in PLA strategy.

NLP	0.3		0.5		0.7		1.0	
LLM	ACC	F1	ACC	F1	ACC	F1	ACC	F1
				CoLA				
Gemini	83.51	0.798	83.31	**0.802**	**83.69**	0.801	82.45	0.791
Llama	**64.11**	0.553	62.73	0.539	63.10	0.542	64.05	**0.579**
Mistral	85.52	0.820	**85.53**	0.820	85.33	0.817	85.04	**0.823**
				CARER				
Gemini	65.75	0.594	65.25	0.586	66.05	0.595	**67.35**	**0.619**
Llama	**64.60**	**0.586**	64.10	0.581	63.05	0.567	64.40	0.583
Mistral	**70.40**	0.636	68.65	0.625	69.70	0.632	70.00	**0.644**

Table 17: A comparison of classification accuracy (%) and F1 score to assess the impact of various trade-off hyperparameters in NLP strategy.