MICL: Improving In-Context Learning through Multiple-Label Words in Demonstration

Anonymous ACL submission

Abstract

In-context learning (ICL) enables large language models (LLMs) to perform new tasks by using sample-label pairs as demonstrations. However, variations in demonstrations can lead to significantly different performances. Current research mainly focuses on selecting demonstration samples, preassuming the class name to be the label word when creating sample-label pairs. However, the choice of label words is crucial for ICL performance. In addition, we observe that using a single class name in demonstration may not yield optimal results. In this paper, we propose to use multiple label words in one sample-label pair to enhance ICL performance. Further, we select and order sample-label pairs based on LLM's output distribution, aiming to optimize the demonstration examples from both the samples' and labels' perspectives. Evaluation results on seven classification datasets show that the use of multiple label words, strategically organized by their selection, order and quantity, improves ICL performance through diverse label information.

1 Introduction

In-context learning (ICL) could perform new tasks by using sample-label pairs from training data as demonstrations, without having to re-train or fine-tune large language models (LLMs) (Brown et al., 2020). The choice of demonstrations is crucial, as ICL performance can vary significantly with different organizations[1] (Liu et al., 2022). To enhance ICL performance, most studies focus on the selection and ranking of samples in demonstrations (Zhang et al., 2022; Hongjin et al., 2022; Levy et al., 2023; Lu et al., 2022), preassuming the class name to be the label word, overlooking the importance of label word selection in demonstrations.

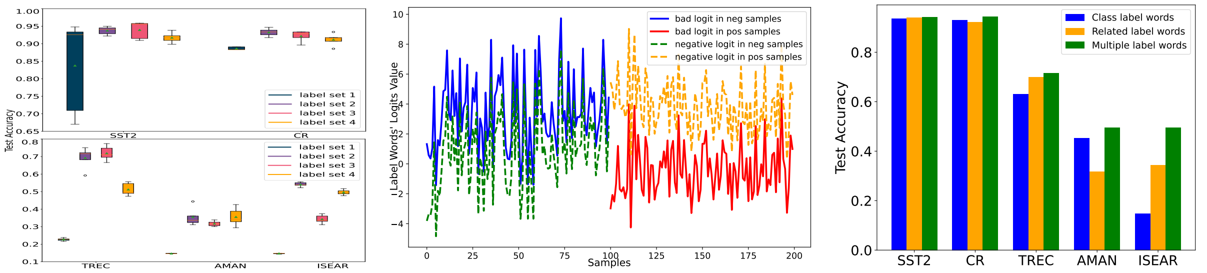
Label words in sample-label pairs may greatly impact ICL performance (Yoo et al., 2022). Fig. 1(a) shows 1-shot ICL performance under varied label words on five datasets, indicating that **carefully selected label words in demonstrations could excel in both accuracy and robustness of ICL**.

LLM's output distribution (logit) over labels is a key consideration in demonstration selection (Rubin et al., 2022) and class prediction (Wang et al., 2023). However, we find that certain class-related words, named as label words in this paper, may fit the LLM better than predefined class names. Fig. 1(b) shows the logit distribution of the label word "bad" and the class name "negative" under the zero-shot setting across negative and positive samples in a sentiment analysis dataset. Obviously, the label word "bad" exhibits a greater difference in logit for samples of the positive and negative classes than the class name "negative". This suggests that **the commonly used class names may not be the best choice for label words in sample-label pair demonstrations for ICL.**

The logits of label words vary significantly across samples. We found that only one label word may be insufficient to express the semantics of the class name. Expanding from a single class name to multiple label words, at the linguistic level, can reduce ambiguity and enrich the semantics of the label name, potentially leading to improved performance. As shown in Fig. 1(c), **the use of multiple label words in sample-label pairs indeed improves ICL performance.**

Verbalizer, which links words related to class names to label space, is beneficial for prompt-based learning (Gao et al., 2021). However, directly employing all the label words in the verbalizer for ICL is infeasible. Firstly, verbalizer often contains hundreds of label words (Hu et al., 2022), and inserting all these label words into ICL prompts can overwhelm the model or exceed token limits. Secondly, ICL relies on demonstration organization. Predefined label words from verbalizers may not fit well with the flexible nature of ICL. Without explicit

[1] In this paper, we follow Wu et al. (2023) to denote the selection and ranking of sample-label pairs as organization.



(a) Label effectiveness in ICL

(b) Label words output distribution in ZSL (more word evaluations are in Appx. A.2)

(c) Multiple label words effectiveness in ICL

Figure 1: Exploration of label words in Llama2-7b. (a): We evaluate four sets of label words under the same samples across five datasets. Long bars indicate instability between seeds, while jumps between bars show accuracy differences among label sets. (b): We evaluate the logit values (y-axis) of various label words on SST-2 samples (x-axis) under zero-shot learning, showcasing 100 negative and positive samples per label word to demonstrate the logit separability of samples to label words. (c): In 1-shot ICL using class names, label-related words, and multiple label words (combining the two sets with spaces) as labels, performance with multiple label words surpassed the other two sets. (a), (b), (c) detailed analysis and experimental settings, including those on GPT2-xl, are in Appx. A.

instructions on sample pairing and order, improper sample-label pairs could mislead the model's understanding. Besides, the impact of employing additional label words in prediction within ICL differs from that in prompt-based methods (Sec. 4.4).

In this work, we propose a new algorithm that selects and orders multiple label words for sample-label pairing based on LLM's output distribution (logit) over training samples. We first filter the related label words collected from the large knowledge base to ensure they are tailored for LLM and dataset under study. We then apply zero-shot learning to training samples to obtain the LLM's output distribution (logit) of the label words. We initialize the label in sample-label pairs using the word with the highest logit value, and then iteratively select and add extra label words to the sample-label pairs. The number of label words added to the sample-label pairs is determined in terms of ICL performance. To improve demonstration organization, we further select and rank samples based on the output distribution of their semantically-related label words and design their corresponding multiple label words.

We summarize our contributions as follows:

1. We propose **MICL**, a method that uses **M**ultiple label words to enhance **ICL**. We develop an algorithm to filter related label words via samples' output distribution, aiming to find label words that best suit the LLM and the data. By using multiple label words in sample-label pairing, more comprehensive label information is provided for ICL, which improves clarity and reduces ambiguity, and this in turn enhances ICL performance.
2. Based on the selected label words' output distribution of training samples, we developed sample organization algorithm, involving selection and ordering of samples for demonstrations, which further improves ICL performance.
3. Extensive experiment results across various classification datasets prove MICL's effectiveness.

2 Related Work

Demonstration organization in ICL To improve ICL performance with better demonstration organization, some studies use pre-trained models like S-BERT (Liu et al., 2022) or BM25 (Hongjin et al., 2022; Levy et al., 2023) to select and rank demonstrations. While these unsupervised methods have advantages, they may cause inconsistencies in knowledge transfer. Other approaches organize demonstration based on the LM's output distribution. Some methods take part of the training set as validation to enable supervised learning methods for demonstration organization (Chang and Jia, 2023; Zhang et al., 2022; Wang et al., 2024). However, this will shrink the pool of candidates, risking sub-optimal selection. Additionally, some methods use the LM's output, like label confidence, to organize demonstrations under the full training set (Rubin et al., 2022; Wu et al., 2023; Li and Qiu, 2023). In the above works, the class names are preassumed to be the label word when creating the demonstration sample-label pairs.

Label words matter in ICL The significance of label words in ICL has been debated. Yoo et al.

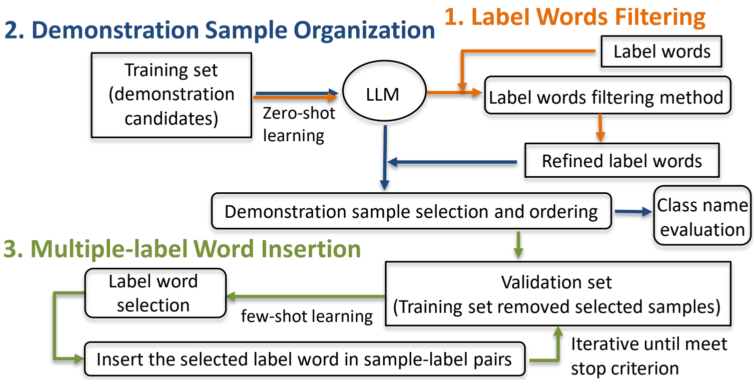

2. Demonstration Sample Organization
1. Label Words Filtering
Label words
Training set (demonstration candidates)
LLM
Label words filtering method
Zero-shot learning
Refined label words
Demonstration sample selection and ordering
Class name evaluation
3. Multiple-label Word Insertion
Label word selection
Validation set (Training set removed selected samples)
few-shot learning
Insert the selected label word in sample-label pairs
Iterative until meet stop criterion

Figure 2: Overall architecture of MICL: Orange lines show Label Words Filtering workflow, blue lines represent Demonstration Sample Organization, and green lines depict Multiple-label Word Insertion.

(2022) demonstrates a positive relationship between ICL performance and accurate sample-label mapping. Li and Qiu (2023) reveals that using the same sample but different labels in the demonstrations can result in very different ICL performance. Wang et al. (2023) suggests that label words derive semantic representations from demonstrations for use in deep layers to make final predictions. Yu and Ananiadou (2024) further shows that these demonstration features are integrated into corresponding labels, with each in-context head extracting features specific to these labels. Milios et al. (2023) uses different label words in augmented samples to enhance ICL performance, indicating the potential for enhanced demonstration effectiveness when multiple label words are used. However, this approach significantly increases the required token length and running time in augmentation settings.

3 Method

In this section, we introduce our proposed method **MICL**, which comprises three parts: label words filtering, demonstration sample organization, and multiple-label word insertion, as shown in Fig. 2.

3.1 Problem Statement

Give a large language model M, class names (label space) L, a label words set S with words related to class names, test sample x_{test} and demonstrations $\{x_i, y_i\}_{i=1}^{L}$[2].

The zero-shot classification of x_{test} can be based on the logits of class names only as $\arg\max_{y\in\{L\}} p_M(y \mid x_{\text{test}})$ or based on the logits of class names and all label words as $\arg\max_{y\in\{L,S\}} p_M(y \mid x_{\text{test}})$. Similarly, the 1-shot prediction of x_{test} is $\arg\max_{y\in\{L\}} p_M(y \mid x_1 \oplus y_1, \cdots, x_L \oplus y_L \oplus x_{\text{test}})$ or $\arg\max_{y\in\{L,S\}} p_M(y \mid x_1 \oplus y_1, \cdots, x_L \oplus y_L \oplus x_{\text{test}})$. $\oplus$ is the concatenation operation, and function $p_M(\cdot)$ returns the logits of words in M's vocabulary. In the following sections, the prediction based on the logits of class names L is referred to as **class-name result**, and the prediction based on the logits of all the label words S and class names L is referred to as **label-words result**.

As analyzed in Section 1, the use of class names only in the sample-label pair $x_i \oplus y_i$ may not be sufficient. In this study, we propose to use multiple label words S_i to create sample-label pair $x_i \oplus S_i$, where S_i is a sequence of multiple label words for class i. The selection and ordering of label words in S_i will be introduced next.

3.2 Label Words Filtering

Given a set of label words relating to a class name, we developed a two-stage filtering algorithm to refine words, tailoring them for the LLM employed and the dataset under study.

Stage 1: Filtering Based on Separability of Logit Distribution Stage 1 filters out words based on the LLM's logit distribution (Alg. 1, lines 3-14). We evaluate each label word's separability based on its logit value across the training set under the zero-shot setting, where the input samples are formatted in a template without labels. The label word filtering is based on the following two principles: (1) label words whose logit values are not the highest for their own class's training samples are discarded (Alg. 1, lines 7-9). These label words do not fit the samples of their own class, providing confusing information and negatively impacting the performance of ICL. (2) Retained words must have the

[2] Our work focus on ICL with label-balanced demonstrations.

maximum average non-negative logit values (Alg. 1, lines 10-12) to ensure semantic representativeness in LLM's feedback. We observed that principle (1) is met across all datasets. Principle (2) is not satisfied in some datasets, so only principle (1) is applied to these datasets in the experiments.

Stage 2: Filtering Based on Point-Biserial Testing In Stage 2, the retained label words are further evaluated by Point-Biserial correlation testing (Tate, 1954). For a label word, if its mean logit value over training samples of its own class differs significantly from that of other classes, it will be retained, otherwise, it will be deleted from the label words set (Alg. 1, lines 15-22). The input in the testing is the logit vector $B \in \mathbb{R}^{1\times|D^l|}$ obtained from 0-shot learning in D^l (Alg. 1, line 4). The label of B is 0 if l differs from the word's label or 1 if they are the same. Significance testing is employed on the correlation results to ensure reliability.

Algorithm 1 Label Words Filtering

```
1: Input: Label set $S$, class names $L$, Train set $D$
2: Output: Refined label set $S_r$
3: Step 1: Distribution Separability Filtering
4: Perform zero-shot learning on $D$, split into sub-train sets
   $D^l$ for $l \in L$
5: for each word $w \in S$ with label $l \in L$ do
6:     Obtain average logit $b_l$ for $w$ in each $D^l$
7:     if label in $\max(\bigcup_{l \in L} b_l) \neq l$ then
8:         Filter out $w$
9:     end if
10:    if $\max(\bigcup_{l \in L} b_l) < 0$ then
11:        Filter out $w$
12:    end if
13: end for
14: Return Refined label set $S_1$
15: Step 2: Point-Biserial Testing Filtering
16: for each word $w \in S_1$ do
17:     Compute Point-Biserial correlation $r$ for logit vector
        $B$ of $w$ in $D$
18:     if $r < 0$ or ($r > 0$ and $p$-value $> 0.05$) then
19:         Filter out $w$
20:     end if
21: end for
22: Return Refined label set $S_r$
```

3.3 Demonstration Sample Organization

After filtering, the remaining label words are semantically aligned with the dataset and LLM. Next, we solve the sample organization problem, including the selection of samples from each class and ordering of the samples in the demonstrations. Sample organization is based on the training samples' LLM output distribution (logit) over the remaining label words S_r. For each sample (prompted in a template without label) in zero-shot learning, the LLM outputs the logit values for all words in S_r obtained in Section 3.2. We rank the words based on their logits. We denote the number of words in S_r belonging to class l as N_l. With two requirements for the selected samples: (1) the words with correct sample-label semantic mappings have higher logits, and (2) to have these words ranked as high as possible within the top N_l, we employ two scoring methods, depending on how well the dataset meets the filtering principles.

Scoring method for datasets meeting both principles For sample t_j in training set D with label l, its score as a demonstration sample is the sum of the top-N_l logit value in S_r (Eq 1). If word w_i does not belong to class l, we set logit $b_{w_i} = 0$.

$$score_{t_j} = \sum_{i=0}^{N_l} (\mathrm{b}_{w_i} * \mathbf{1}_{l_{t_j}=l_{w_i}}) \qquad (1)$$

Scoring method for datasets meeting principle 1 Since the logit as a feature for selection isn't representative (with negative values), we score the sample t_j based on the top-N_l linear weighted ranking position score of the words in S_r instead of the logit value as Eq 2. We set its ranking position score to zero if w_i does not belong to class l.

$$score_{t_j} = \sum_{i=0}^{N_l} \left(\frac{2(N_l - i)}{(N_l + 1)N_l} * \mathbf{1}_{l_{t_j}=l_{w_i}} \right) \qquad (2)$$

In the k-shot ICL, for each class l, the samples with the top-k scores in D_l are selected as the demonstration sample for class l. The order of the maximum scores in each label determines the demonstration k-shot order.

Class name evaluation To access the semantic information represented by labels in the demonstrations, we analyze logit values of class names in selected samples after selection. We discard the class name l if its logit value, obtained from samples labeled l, ranks in the bottom half of N_l. We then replace it with the label word with top-1 logit to form the initial sample-label pair.

3.4 Multiple-Label Words Insertion in Sample-Label Pairs

We employ multiple label words in sample-label to provide diverse label semantics prompting. Our method sequentially inserts multiple label words into a single sample-label pair, which is more efficient in computation and memory than using multiple augmented sample-label pairs (Milios et al., 2023).

The validation set D_{dev} is the train set D with the demonstration samples removed. D^l_{dev} is the subset of D_{dev} containing all samples of class l. D^l_{dev} is evaluated under the initial sample-label pairs obtained from Sec. 3.3 and outputs the logit value of w of class l in S_r (excluding words in sample-label pairs). For each label name $l \in L$, we pick the one with the maximum average logit in D^l_{dev}, and insert it to form sample-multiple-label pairs. This insertion is iterative, with updated demonstrations at every iteration, until no candidates are left in S_r or the insertion of the additional label words results in a degraded performance on the validation set.

4 Experiments

In this section, we examine the capacities of multiple label words in ICL from five perspectives: (1) 1-shot ICL classification performance with/without multiple label words insertion in baseline models and MICL (Sec. 4.2); (2) 5-shot ICL (Sec. 4.3); (3) The impact of leveraging extra label word mappings in prediction with/without it appearing in demonstrations (Sec. 4.4); (4) The effectiveness of MICL demonstration order compared to enumerating other permutations in ICL (Sec. 4.5); and (5) The influence of MICL with/without label-balanced demonstration (Sec. 4.6).

4.1 Setups

Datasets We evaluate our method on seven datasets, including SST-2 (Socher et al., 2013), CR (Ding et al., 2008), IMDB (Maas et al., 2011), ISEAR (Scherer and Wallbott, 1994), AMAN (Aman and Szpakowicz, 2008), TREC-6 (Li and Roth, 2002) and AGNews (Zhang et al., 2015). We adopt the templates in Wang et al. (2023). The initial label word sets are from Hu et al. (2022) and Zhu et al. (2024). Detailed information on the datasets, templates and label sets is given in Appx. B.

Experiment Settings We conduct few-shot learning experiments using Llama2-7b (Touvron et al., 2023) and GPT2-xl (Radford et al., 2019) to test the effectiveness of our methods.

For the baseline models, including **vanilla Llama2-7b**, **vanilla GPT2-xl**, **Topk** (Liu et al., 2022), **SelfICL** (Wu et al., 2023), and **DataICL** (Chang and Jia, 2023), we adopt their proposed demonstration samples and paired them with class name and our multiple label words, respectively, to create sample-label pairs. Detailed information on the baseline models and their experiment settings is provided in Appx. C.

For each model, multiple-label word insertion is evaluated on validation set, which is the full training set excluding the samples selected for demonstrations. In Vanilla, TopK, and SelfICL, the full training set is split into a reduced training set and a validation set in an 8:2 ratio as demonstration samples vary with test samples. The demonstration samples are selected based on reduced training set, while the multiple-label word insertion is based on validation set. The ICL performance is evaluated on the same test set for all experiments. Results using class names in demonstrations are predicted based on the maximum logits over class names.

4.2 Main Results

We next present the results of each model and MICL (ours) under two settings: with and without multiple-label word insertion in 1-shot ICL. The label words in the initial sample-label pairs are updated label words in MICL and DataICL, while other models use class names. The best test accuracy achieved with multiple-label words inserted in demonstrations before meeting the stopping criterion (validation accuracy decreases with the insertion of label words) is reported as the result of '+Demo-MLabels'. Fig. 3 shows validation and test performance across different numbers of inserted label words (N) for each dataset in MICL. Details on inserted word settings in baseline models and ours for 1-shot settings are given in Appx. D.

The results in Table 1 lead to two conclusions:

1. Integrating multiple label words in demonstrations significantly enhances ICL performance. Across all baseline models, the use of multiple label words led to an average accuracy improvement of over 2% in Llama2-7b (with a maximum of 2.60% in SelfICL) and over 5% in GPT2-xl (with a maximum of 7.37% in vanilla-GPT2-xl).

2. MICL outperforms baselines across all datasets under the initial label word. It achieves an average accuracy improvement of 3.11% in Llama2-7b and 11.58% in GPT2-xl. Notably, the improvement in multi-class classification tasks is more significant: with accuracy gains of 6.02% (AMAN, Llama2-7b), 12.80% (TREC, GPT2-xl), 12.53% (AMAN, GPT2-xl), and 23.68% (AGNews, GPT2-xl).

By using multiple label words, MICL has an average accuracy improvement of 2.33% in Llama2-7b (with a maximum of 9.60% in AMAN) and 3.84% in GPT2-xl (with a maximum of 11.09% in ISEAR) compared with the initial setting where a

	SST2	CR	IMDB	TREC	AMAN	ISEAR	AGNews	Avg.
Llama2-7b								
vanilla-Llama2-7b	93.06	93.24	94.81	68.20	53.08	70.91	81.78	79.30
+Demo-MLabels_CN	93.74↑	94.41↑	95.89↑	69.88↑	59.75↑	71.48↑	83.36↑	81.22
+Demo-MLabels_LW	93.74	94.41	95.89	72.88↑	59.05	71.03	83.88↑	81.55
TopK	92.37	92.82	94.29	77.80	51.38	64.32	79.09	78.87
+Demo-MLabels_CN	93.52↑	93.55↑	94.69↑	83.20↑	58.15↑	67.38↑	79.59↑	81.44
+Demo-MLabels_LW	93.52	93.55	94.69	83.20	58.65↑	66.98	79.59	81.45
SelfICL	91.71	93.35	94.69	76.00	53.63	65.18	81.46	79.36
+Demo-MLabels_CN	92.53↑	92.82↑	95.10↑	79.00↑	58.65↑	69.10↑	82.01↑	81.39
+Demo-MLabels_LW	92.53	92.82	95.10	**82.80**↑	58.65	69.30↑	82.01	81.96
DataICL	94.51	89.89	94.60	71.80	53.88	70.17	83.45	79.76
+Demo-MLabels_CN	95.28↑	92.55↑	94.70↑	74.40↑	55.64↑	71.30↑	84.83↑	81.40
+Demo-MLabels_LW	95.28	92.55	94.70	78.80↑	54.89	71.10	85.45↑	81.95
MICL	95.39^{1}	94.41^{1}	95.40^{1}	78.40^{1}	59.90^{2}	72.49^{2}	84.11^{2}	82.87
+Demo-MLabels_CN	95.97^{1}↑	95.15^{1}↑	95.60^{1}↑	79.80^{1}↑	65.16^{2}↑	**73.55**^{2}↑	86.55^{2}↑	84.54
+Demo-MLabels_LW	**95.97**^{1}	**95.15**^{1}	**95.60**^{1}	80.60^{1}↑	**69.40**^{2}↑	73.09^{2}	**86.58**^{2}↑	**85.20**
GPT2-xl								
vanilla-GPT2-xl	71.74	64.26	67.00	46.84	29.97	39.00	55.24	53.44
+Demo-MLabels_CN	85.63↑	**67.07**↑	68.23↑	53.60↑	39.35↑	50.01↑	58.44↑	60.33
+Demo-MLabels_LW	85.63	67.07	70.13↑	54.76↑	39.30	50.56↑	58.24	60.81
TopK	69.41	65.69	63.36	56.20	32.83	44.12	54.33	55.13
+Demo-MLabels_CN	84.51↑	66.22↑	65.47↑	60.80↑	40.85↑	54.75↑	55.24↑	61.12
+Demo-MLabels_LW	84.51	66.22	67.07↑	61.80↑	40.60	54.88↑	55.24	61.47
SelfICL	70.07	64.89	60.96	56.00	32.58	44.98	54.50	54.85
+Demo-MLabels_CN	83.80↑	64.89	61.46↑	64.20↑	43.36↑	54.82↑	56.42↑	61.28
+Demo-MLabels_LW	83.80	64.89	62.16↑	65.40↑	43.11	55.08↑	56.42	61.55
DataICL	83.47	63.83	64.80	57.20	35.34	35.28	43.36	54.75
+Demo-MLabels_CN	84.84↑	63.83	69.80↑	58.20↑	36.84↑	48.04↑	51.14↑	58.96
+Demo-MLabels_LW	84.84	63.83	69.20	65.20↑	36.84	48.04	51.14	59.87
MICL	85.17^{2}	64.89^{1}	71.50^{1}	70.00^{2}	47.87^{2}	48.64^{2}	78.92^{2}	66.71
+Demo-MLabels_CN	91.65^{2}↑	65.96^{1}↑	73.40^{1}↑	70.40^{2}↑	49.62^{2}↑	**59.73**^{2}↑	79.08^{2}↑	**70.55**
+Demo-MLabels_LW	**91.65**^{2}	65.96^{1}	**73.40**^{1}	**70.40**^{2}	**49.87**^{2}↑	58.74^{2}	**79.49**^{2}↑	70.50

Table 1: ICL Experimental Results: '+Demo-MLabels_CN' refers to the class-name result with multiple-label words enhanced in the demonstration (predicted based on the maximum logits over class names), and '+Demo-MLabels_LW' refers to the label-words result with multiple-label words enhanced in the demonstration (predicted based on the maximum logits over inserted label words). The best accuracy results (%) are marked in bold. Marker ^{1} indicates results given under Eq 1, while marker ^{2} indicates results given under Eq 2. Upward arrow (↑) in '+Demo-MLabels_CN' signifies an increase in performance compared to the original method, while in '+Demo-MLabels_LW', it signifies an increase in performance compared to '+Demo-MLabels_CN'.

single label word is used in sample-label pairing.

Enhancement via initial label words updates To ensure the quality of initial sample-label pairs, we update certain labels in two datasets based on the samples selected in MICL and DataICL before ICL experiments[3]. In AMAN, 'other' is replaced with 'neutral'. In TREC, 'entity' is replaced with 'animal', 'description' with 'definition', 'human' with 'persons', 'location' with 'state', and 'number' with 'numeric'. The enhanced accuracy presented in Table 2 demonstrates the superiority of our initial label word replacement method over the direct use of class names for improving 1-shot ICL.

	Llama2-7b				GPT2-xl			
	DataICL		MICL		DataICL		MICL	
	original	update	original	update	original	update	original	update
AMAN	40.85	**53.88**	57.00	**59.90**	31.08	**35.34**	41.35	**47.87**
TREC	70.20	**71.80**	70.40	**78.40**	37.40	**57.20**	54.20	**61.60**

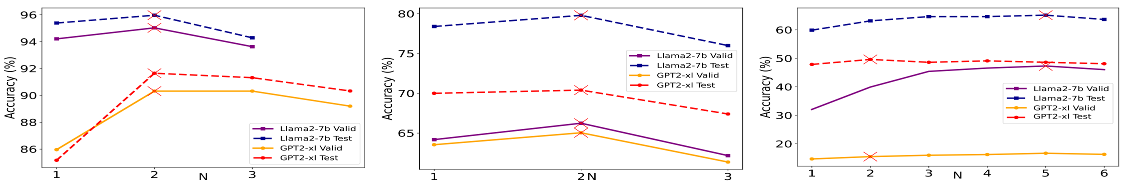
Table 2: Enhancement in ICL accuracy (%) through label evaluation and updates.

4.3 Effectiveness of MICL in 5-shot ICL

To explore the capability of multiple-label words further, we assess their effectiveness in a 5-shot setting using our method and SelfICL for one binary classification task (SST2) and one multi-class classification task (AMAN). For both methods, the 5-shot sample-label pairs are selected based on the top-5 scoring samples in the training set for each label. The validation set is created by removing all selected samples from the training set. The sample-label pairs are ordered based on the highest score achieved by each label among the selected samples.

As shown in Table 3, the insertion of multiple label words is effective in 5-shot settings, yielding an average accuracy improvement of 3.64% and 9.80% in SelfICL, and 1.55% and 12.49% in MICL under Llama2-7b and GPT2-xl, respectively. MICL outperforms SelfICL across all results. Details on inserted words in baseline models and ours for 5-

[3] Vanilla, TopK, and SelfICL demonstration samples vary from test samples, leading to non-uniform sample-label pairs among the test samples in label evaluation. Consequently, we apply class names as the initial label words in the multiple-label word insertion experiment.



(a) SST-2 (b) TREC (c) AMAN

Figure 3: Validation and test performance under different label word quantity (N) in sample-multiple-label pairs. The red cross marks the reported result setting. In Llama2-7b, N is 2 for SST2, IMDB, TREC, ISEAR, and AGNews, 4 for CR, and 5 for AMAN. In GPT2-xl, N is 7 for ISEAR and 2 for others. The remaining datasets are in Appx. D.

	SST2	AMAN	Avg.	SST2	AMAN	Avg.
		Llama2-7b			*GPT2-xl*	
SelfICL	93.63	50.13	71.88	73.04	34.56	53.80
+Demo-MLabels_CN	94.89↑	55.89↑	75.39	86.60↑	40.35↑	63.48
+Demo-MLabels_LW	94.89	56.14↑	75.52	86.60	40.60↑	63.60
MICL	94.56	56.14	75.35	81.16	34.84	58.00
+Demo-MLabels_CN	95.39↑	58.40↑	76.90	89.18↑	51.80↑	**70.49**
+Demo-MLabels_LW	**95.39**	**58.40**	**76.90**	**89.18**	**51.80**	70.49

Table 3: Multiple-label word insertion accuracy (%) in 5-Shot ICL: Scoring method and test sets as in Table 1.

	SST2	TREC	AMAN	ISEAR	AGNews	Avg.
Llama2-7b						
ZSL	88.96	68.80	48.87	58.60	67.29	66.50
ZSL_Sr	90.94	60.00↓	47.37↓	60.80	60.03↓	63.83
MICL	95.39	78.40	59.90	72.49	78.06	76.85
MICL_Sr	95.44	72.00↓	63.13	71.56↓	74.37↓	75.30
+Demo-MLabels_CN	95.97	79.80	65.16	73.55	86.55	80.21
+Demo-MLabels_LW	95.97	80.60	69.40	73.09	86.58	81.13
+Demo_MLabels_Sr	94.51↓	74.40↓	67.64↓	71.56↓	75.33↓	76.69
GPT2-xl						
ZSL	79.57	38.00	39.60	42.33	53.16	50.53
ZSL_Sr	50.30↓	46.80	37.84↓	42.52	51.29↓	45.75
MICL_C	85.17	61.60	47.87	48.64	78.92	64.44
MICL_Sr	85.17	67.60	47.62↓	48.70	65.28↓	62.87
+Demo-MLabels_CN	91.65	70.40	49.62	59.73	79.08	70.10
+Demo-MLabels_LW	91.65	70.40	49.87	58.74	79.49	70.03
+Demo_MLabels_Sr	91.65	68.80↓	49.87	58.64↓	67.58↓	67.31

Table 4: Impact of prediction under class names, multiple-label words in demonstration and S_r. Arrow ↓ indicates a decrease in accuracy of predictions over S_r, compared to those over class names (CN) or inserted label words (LW).

shot settings are in Appx. D, Table 11.

4.4 Analysis of Utilizing Multiple Label Words Mapping in Prediction

Table 1 and 3 show that '+Demo-MLabels_LW' (prediction based on the maximum logit over inserted label words) achieves higher accuracy than '+Demo-MLabels_CN' (prediction based on the maximum logit over class names) in some datasets. This seems to align with the idea that incorporating extra label word mapping in the final prediction can enhance prompt-based methods (Schick and Schütze, 2021). However, in ICL, models tend to predict based on the labels provided in demonstrations, which may perform differently with extra label word mappings in prediction. We compare the effectiveness of using extra label word mappings only in predictions versus including them in demonstrations for both binary-class and multi-class tasks. Additionally, we evaluate the impact of leveraging extra label word mappings under zero-shot settings.

In Table 4, most experiments show decreased performance when predicting based on label words in S_r compared to predictions on label words that appeared in demonstrations (including class names). This suggests that simply applying extra label word mappings in the final prediction might disrupt information learned from demonstrations in ICL. Surprisingly, even under zero-shot learning (ZSL), where label information isn't prompted, adding extra label knowledge still decreases some classification performance. These results highlight the complex role of labels in ICL classification.

4.5 Effectiveness of MICL Ordering

In MICL, we use LLM's feedback over label words as a decision feature to order sample-label pairs. Although this approach may not provide the optimal ordering compared to evaluating all possible permutations, the computational cost of enumerating all possibilities in multi-class tasks is prohibitive. To assess the effectiveness of our ordering method, we compare the classification performance using MICL's initial sample-label pairs against 30 and 50 random permutations (excluding MICL's order) in multi-class tasks in Llama2-7b and GPT2-xl, respectively, and the flipped order in binary tasks.

As shown in Table 5, MICL often outperforms or matches the best results among compared permutations. It achieves an average accuracy improvement of 0.39% in Llama2-7b and 5.86% in GPT2-xl. Despite a notable performance drop for IMDB (GPT2-xl), the binary task nature can mitigate this by enumerating all orders. MICL excels

	SST2	CR	IMDB	TREC	AMAN	ISEAR	AGNews	Avg.
Llama2-7b								
MICL	**95.39**	**94.41**	**95.40**	**78.40**	59.90	72.49	84.11	**82.87**
Permutation	92.97	93.35	94.40	78.20	**60.15**	72.49	**85.82**	82.48
GPT2-xl								
MICL	**85.17**	**65.96**	71.50	**70.40**	**47.87**	48.64	**78.92**	**66.92**
Permutation	53.87	63.83	**82.20**	63.20	39.10	**52.03**	73.16	61.06

Table 5: Effectiveness of MICL's Order: Comparison of our ordering method with random orders in multi-class datasets and the flipped order in binary datasets. 'Permutation' presents the best result among the evaluated permutations for each dataset.

in multi-class tasks, matching or exceeding top permutation results. This highlights the effectiveness of MICL's demonstration order, particularly in multi-class tasks with thousands of possible orderings (7!), which would take months to process enumerating in LLMs, whereas MICL achieves comparable performance within hours or minutes.

4.6 Effectiveness of Label Balance in MICL

Our method is evaluated under a label-balanced demonstration setting, assuming that every category of label information matters. We also investigate an unbalanced setting by removing a sample-label pair with the highest sample score according to our scoring methods. Table 6 summarizes the results for two language models in 1-shot ICL.

	SST2	CR	IMDB	TREC	AMAN	ISEAR	AGNews	Avg.
Llama2-7b								
MICL	95.39	94.41	95.40	78.40	59.90	72.49	84.11	82.87
MICL unbalanced	91.98	93.35	93.60	68.00	57.39	70.70	81.49	79.50
+Demo-MLabels_CN	95.97	95.15	95.60	79.80	65.16	**73.55**	86.55	84.54
+Demo-MLabels_CN unbalanced	92.42	93.62	94.40	69.60	62.16	71.36	85.87	81.35
+Demo-MLabels_LW	**95.97**	**95.15**	**95.60**	80.60	**69.40**	73.09	**86.58**	**85.20**
+Demo-MLabels_LW unbalanced	92.42	93.62	94.40	81.80	62.91	71.36	86.36	83.27
GPT2-xl								
MICL	85.17	64.89	71.50	61.60	47.87	48.64	78.92	66.71
MICL unbalanced	51.02	**73.67**	67.80	53.60	41.60	54.62	34.45	53.82
+Demo-MLabels_CN	91.65	65.96	73.40	70.40	49.62	**59.73**	79.08	**70.55**
+Demo-MLabels_CN unbalanced	52.85	74.47	68.20	55.40	45.86	56.75	43.75	56.75
+Demo-MLabels_LW	**91.65**	65.96	**73.40**	**70.40**	**49.87**	58.74	**79.49**	70.50
+Demo-MLabels_LW unbalanced	52.85	74.47	68.40	59.60	45.61	56.42	43.75	57.30

Table 6: Accuracy performance (%) of MICL, '+Demo-MLabels_CN', '+Demo-MLabels_LW' under label-balanced and label-unbalanced demonstrations.

The impact of label-unbalanced demonstrations varies between the two models. For Llama2-7b, a large-size language model, the negative effects of missing label information are less marked, with an average accuracy decrease of 3.37%. The most significant accuracy drops in TREC by 10.40%. The leverage of multiple-label words mitigated the performance loss, reducing the average accuracy decline to 1.93% in '+Demo-MLabels_LW' compared to a label-balanced setting. In contrast, GPT2-xl, a smaller model with 1.5 billion parameters, experienced a marked performance decline under unbalanced conditions, averaging a 12.89% decrease in accuracy. Specifically, SST2 and AGNews experienced over 34% declines. Interestingly, CR and ISEAR demonstrated improved performance despite the unbalanced labels. The incorporation of multiple-label words consistently enhanced performance across all datasets in the unbalanced setting, affirming their utility in ICL.

4.7 Label Word Filtering Results

Table 7 lists the number of filtered words (-) for each dataset under Llama2-7b and GPT2-xl during distribution separability filtering (S_1) and Point-Biserial testing filtering (S_r).

Model	Filtered Verbalizer	SST2	CR	IMDB	TREC	AMAN	ISEAR	AGNews
Llama2-7b	S_1	-315	-228	-231	-31	-75	-78	-1163
	S_r	-1	-9	-9	-1	-3	0	0
GPT2-xl	S_1	-393	-368	-383	-40	-62	-118	-645
	S_r	-4	-7	-6	-2	-6	0	-3

Table 7: The statistic information of filtering results under two-stage filtering.

The large number of words filtered by distribution separability indicates that although many words match the task topic definition at the linguistic level, they are not suitable as label words at the LLM level. This suggests that simple label-based voting, commonly used in many prompt-based methods, might harm LLM's in-context learning, as the candidate words do not align with the task based on the LLM's understanding. The small number of words filtered by Point-Biserial testing indicates the high quality of the proposed distribution separability filtering, as the remaining words show significant separation between the true class and false classes based on logit vectors.

5 Conclusion

This paper introduces MICL, a novel approach to organizing demonstrations with multiple-label words inserted in in-context learning (ICL). By utilizing a variety of label words and analyzing their distribution within large language models (LLMs), we enhance ICL understanding by providing diverse label information. We develop a structured method for selecting and ordering sample-multiple-label pairs via LLM's feedback over label words. Extensive experimental results show that our method of multiple-label word insertion significantly improves ICL classification performance, yielding superior results.

6 Limitations

In this paper, we enhance in-context learning performance by incorporating additional label-related words. Although related label words for various tasks have been extracted and collected, new datasets may still lack appropriate label words. However, powerful search tools such as WordNet (Pedersen et al., 2004), ConceptNet (Speer et al., 2017), and open-source vocabularies can mitigate this issue. Our method designs a filtering approach that refines the quality of label words based on these search results.

References

Saima Aman and Stan Szpakowicz. 2008. Using roget's thesaurus for fine-grained emotion recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

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.

Ting-Yun Chang and Robin Jia. 2023. Data curation alone can stabilize in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8123–8144.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

SU Hongjin, 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. In *The Eleventh International Conference on Learning Representations*.

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

Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1401–1422, Toronto, Canada. Association for Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235.

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, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. 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*, pages 100–114.

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, Dublin, Ireland. Association for Computational Linguistics.

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

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.

Ted Pedersen, Siddharth Patwardhan, Jason Michelizzi, et al. 2004. Wordnet:: Similarity-measuring the relatedness of concepts. In *AAAI*, volume 4, pages 25–29.

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.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671.

Klaus R Scherer and Harald G Wallbott. 1994. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66(2):310.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

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

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Robert F Tate. 1954. Correlation between a discrete and a continuous variable. point-biserial correlation. *The Annals of mathematical statistics*, 25(3):603–607.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2024. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 36.

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 *The 61st Annual Meeting of the Association for Computational Linguistics*.

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-Goo Lee, and Taeuk Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. In *2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, pages 2422–2437. Association for Computational Linguistics (ACL).

Zeping Yu and Sophia Ananiadou. 2024. How do large language models learn in-context? query and key matrices of in-context heads are two towers for metric learning. *arXiv preprint arXiv:2402.02872*.

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

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148.

Zixiao Zhu, Junlang Qian, Zijian Feng, Hanzhang Zhou, and Kezhi Mao. 2024. EDEntail: An entailment-based few-shot text classification with extensional definition. In *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

A Appendix A

A.1 Label Effectiveness in ICL

This study establishes four distinct label sets for each dataset, utilizing identical samples to form the sample-label pairs in 1-shot ICL, as detailed in Table 8. The reported results represent the average accuracy obtained from five repeated experiments, conducted with seeds 42, 43, 44, 45, and 46 in sample selection during the 1-shot demonstrations. In Llama2-7b, the maximum 49.20% accuracy difference (TREC), and the highest 11.08% (SST2) standard deviation are observed. We also evaluate the label effectiveness in GPT2-xl under the same experimental conditions, with results shown in Fig. 4. The results are similar to those in Llama2-7b, with a maximum accuracy difference of 48.20% (TREC) and a maximum standard deviation of 10.52% (SST2). These findings indicate that label selection contributes to both ICL accuracy and robustness.

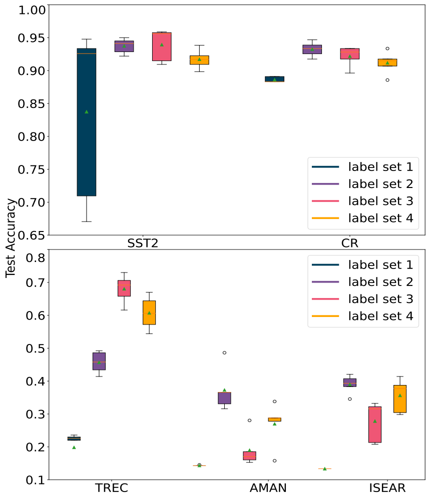


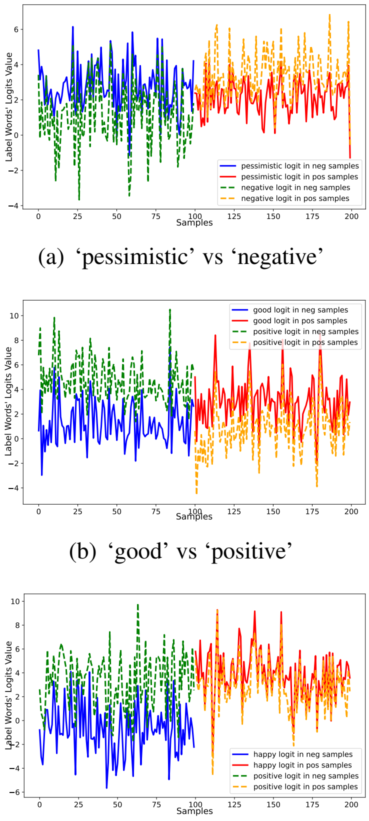
Figure 4: Label effectiveness in ICL (GPT2-xl)

Label Set	SST2	CR	TREC	AMAN	ISEAR
1	0: ‘0’, 1: ‘1’	0: ‘0’, 1: ‘1’	0: ‘0’, 1: ‘1’, 2: ‘2’, 3: ‘3’, 4: ‘4’	0: ‘0’, 1: ‘1’, 2: ‘2’, 3: ‘3’, 4: ‘4’, 5: ‘5’, 6: ‘6’	0: ‘0’, 1: ‘1’, 2: ‘2’, 3: ‘3’, 4: ‘4’, 5: ‘5’, 6: ‘6’
2	0: ‘ negative’, 1: ‘ positive’	0: ‘ negative’, 1: ‘ positive’	0: ‘ abbreviation’, 1: ‘ entity’, 2: ‘ description’, 3: ‘ human’, 4: ‘ location’,5: ‘location’	0: ‘ fear’, 1: ‘ sadness’, 2: ‘ disgust’, 3: ‘ anger’, 4: ‘ joy’, 5: ‘ surprise’, 6: ‘ others’	0: ‘ fear’, 1: ‘ sadness’, 2: ‘ disgust’, 3: ‘ anger’, 4: ‘ joy’, 5: ‘ guilt’, 6: ‘ shame’
3	0: ‘ bad’, 1: ‘ good’	0: ‘ bad’, 1: ‘ good’	0: ‘ abbreviation’, 1: ‘ animal’, 2: ‘ definition’, 3: ‘ persons’, 4: ‘ state’,5: ‘ numeric’	0: ‘ worry’, 1: ‘ sadness’, 2: ‘ loathing’, 3: ‘ rage’, 4: ‘ happy’, 5: ‘ stunning’, 6: ‘ neutral’	0: ‘ worry’, 1: ‘ grief’, 2: ‘ loathing’, 3: ‘ rage’, 4: ‘ happy’, 5: ‘ remorse’, 6: ‘ embarrassment’
4	0: ‘ terrible’, 1: ‘ great’	0: ‘ terrible’, 1: ‘ great’	0: ‘ abbreviation’, 1: ‘ food’, 2: ‘ reason’, 3: ‘ persons’, 4: ‘ city’, 5: ‘ count’	0: ‘ anxiety’, 1: ‘ sad’, 2: ‘ disgusting’, 3: ‘ angry’, 4: ‘ pleasure’, 5: ‘ surprising’, 6: ‘ noemo’	0: ‘ anxiety’, 1: ‘ sad’, 2: ‘ disgusting’, 3: ‘ angry’, 4: ‘ pleasure’, 5: ‘ regret’, 6: ‘ humiliation’

Table 8: The label information of each dataset.

A.2 LLM’s Output Separability Over Label Words in Zero-Shot Learning

This study evaluates the logits value separability for negative and positive samples of label words ‘bad’ and ‘pessimistic’ compared to the class name ‘negative’, and for label words ‘good’ and ‘happy’ compared to the class name ‘positive’ in SST-2 under zero-shot learning. Except for the word bad’, shown in Fig. 1(b), the logit distribution figures for the remaining words are listed in Fig 5.



(a) ‘pessimistic’ vs ‘negative’

(b) ‘good’ vs ‘positive’

(c) ‘happy’ vs ‘positive’

Figure 5: Label words logit separability over samples.

In Fig. 1(b) and Fig. 5, the first 100 samples are negative, while samples 101-200 are positive samples. Compared to the class names, the logits of negative label words across negative samples are higher than those for ‘negative’, while the logits for the same label words across positive samples are lower than ‘negative’. Similarly, the logits for positive label words are higher across positive samples and lower across negative samples than ‘positive’, indicating better logit separability for these label words compared to their respective class names. This demonstrates superior logit separability for certain label words compared to class names. Since logit values are crucial for class prediction, this enhanced separability can significantly improve classification performance.

A.3 Multiple label words effectiveness in ICL

This study evaluates the performance of demonstrations using different numbers of label words. The multiple-label words combine the class name with related label words, connected by spaces, such as "negative bad" and "positive good" in SST2 and CR. We also assess the label effectiveness in GPT2-xl under the same experimental conditions, with the results shown in Fig. 6. These findings indicate the potential of using multiple-label words in demonstrations to enhance ICL.

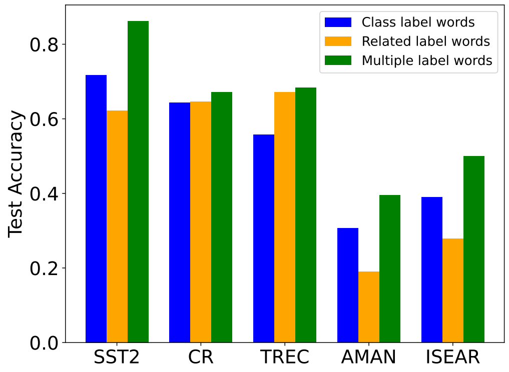


Figure 6: Multiple label words effectiveness in ICL (GPT2-xl)

B Appendix B

The Statistic Information of Datasets and Templates are listed in Table 9. Suppose the original dataset has no train/test split. In that case, a testing set is randomly selected, comprising 20% of the entire dataset with a balanced-label distribution, while the remaining data is used for training (AMAN, ISEAR). If the dataset includes a validation set, the original training and validation sets are combined to form a complete training set for demonstration selection (SST2). For AGNews, only 4,000 training samples are selected, with 1,000 samples per label, due to memory constraints.

The label word sets for SST2, CR, IMDB, and AGNews are derived from Hu et al. (2022), while those for TREC, AMAN, and ISEAR are sourced from Zhu et al. (2024).

Dataset	Template	Class Name	#Train	#Validation	#Test
SST2	Review: Sentiment:	positive, negative	6920	872	1821
CR	Review: Sentiment:	positive, negative	3394	-	377
IMDB	Review: Sentiment:	positive, negative	1000	-	1000
AMAN	Review: Emotion:	angry, disgust, joy, others, surprise, sad, and fear	4090	-	-
ISEAR	Review: Emotion:	angry, disgust, joy, shame, guilt, sadness, and fear	7666	-	-
TREC	Question: Answer Type:	location, number, description, entity, human, and abbreviation	5451	-	490
AGNews	Article: Answer:	Worlds, Business, Sports, and Technology	120000	-	7600

Table 9: The applied template and statistic information in each dataset.

C Appendix C

All experiments are implemented under Python 3.8 environment and PyTorch 2.1.0. with Cuda version 11.8, GPU NVIDIA RTX A5000.

Baseline Model Experimental Settings The detailed information on the baseline models and the corresponding experimental settings for few-shot learning experiments is provided below.

Vanilla Llama2-7b (Touvron et al., 2023): We use Llama2-7b[4], a 7 billion parameter language model with 4096 tokens available. Prompts exceeding the model's token limit are truncated in the few-shot settings. The demonstrations are randomly selected and ordered on each label using five random seeds: 42, 43, 44, 45, and 46. The reported results are the average ICL accuracy over five runs.

Vanilla GPT2-xl (Radford et al., 2019): We use GPT2-xl[5], a 1.5 billion parameter language model with 1024 tokens available. Prompts exceeding the model's token limit are truncated. The demonstration and results settings are the same as Vanilla Llama2-7b.

TopK (Liu et al., 2022): An unsupervised method selects the nearest neighbors of the test samples as the demonstration samples using S-BERT[6]. In the re-run experiment, we choose samples for each label in the order ranked by their semantic similarity to the test sample.

SelfICL (Wu et al., 2023): A supervised method selects demonstration samples via S-BERT and ranks them based on Minimum Description Length (MDL). In the re-run experiment, after selecting the candidates, we randomly choose 30 combinations (the default setting) containing one sample for each label for MDL ranking with a window size 10. The best results are used as the selected-and-ranked demonstrations for ICL testing.

DataICL (Chang and Jia, 2023): A supervised method trains a linear regressor to fit the LLM's output based on which sample is present and its order in the demonstration. In the re-run experiment, we select the sample with the highest score in each label as the demonstration samples, following the resulting order. The LLM used in DataICL is the same as the ICL evaluation model.

For all experiments conducted with Llama2-7b, the model is configured to operate under a 4-bit setting.

D Appendix D

The Number of Inserted Labels Settings The number of label words (N) used in Table 1 on the baseline models are summarized in Table 10.

	SST2	CR	IMDB	TREC	AMAN	ISEAR	AGNews
Llama2-7b							
vanilla-Llama2-7b	2	5	4	2	5	6	5
TopK	2	2	2	2	6	2	2
SelfICL	2	2	2	2	4	3	2
DataICL	2	2	2	2	3	6	6
MICL	2	4	2	2	5	2	2
GPT2-xl							
vanilla-GPT2-xl	3	2	2	4	3	6	2
TopK	3	3	2	4	6	5	2
SelfICL	3	2	2	3	6	5	2
DataICL	2	2	2	2	2	4	2
MICL	2	2	2	2	2	7	2

Table 10: The number of label words (N) inserted in demonstration in the baseline models and MICL (ours) under each dataset in 1-shot ICL.

The remaining datasets' validation and test accuracy performance like Fig. 3 is shown in Fig. 7.

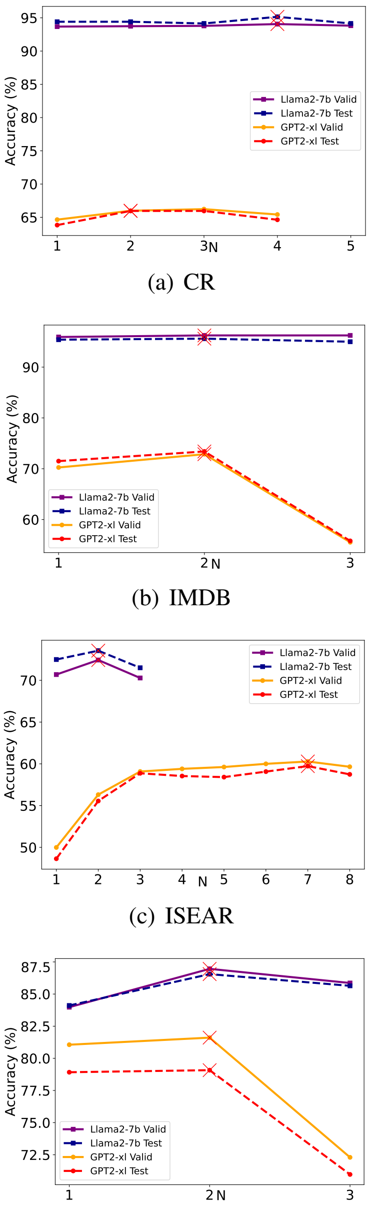
The number of label words (N) used in Table 3 on the baseline models are summarized in Table 11.

For simple classification tasks (binary tasks) such as SST2, CR, and IMDB, inserting around

[4] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

[5] https://huggingface.co/openai-community/gpt2-xl

[6] https://huggingface.co/sentence-transformers/all-mpnet-base-v2



(a) CR

(b) IMDB

(c) ISEAR

(d) AGNews

Figure 7: Validation and test performance under different label word quantity (N) in sample-multiple-label pairs for CR. IMDB, ISEAR and AGNews.

	SST2	AMAN	SST2	AMAN
	Llama2-7b		*GPT2-xl*	
SelfICL	2	6	4	4
MICL	2	3	3	5

Table 11: The number of label words (N) inserted in demonstration in SelfICL and MICL (ours) under SST2 and AMAN in 5-shot ICL.

2 to 3 label words yields good performance. In contrast, for fine-grained tasks (multi-class tasks) such as AMAN, ISEAR, and AGNews, inserting more label words is necessary to achieve better performance. Additionally, the larger language model (Llama2-7b) can effectively handle more label information compared to the smaller language model (GPT2-xl).