ICL-Bandit: Relevance Labeling in Advertisement Recommendation Systems via LLM

Anonymous ACL submission

Abstract

Measuring the relevance between user queries and advertisements is a critical task for advertisement (ad) recommendation systems, such as Microsoft Bing Ads and Google Ads. Traditionally, this requires expert data labeling, which is both costly and time-consuming. Recent advances have explored using Large Language Models (LLMs) for labeling, but these models often lack domain-specific knowledge. In-context learning (ICL), which involves providing a few demonstrations, is a common practice to enhance LLM performance on domain-specific tasks. However, retrieving high-quality demonstrations in a vast exploration space remains challenging. In this paper, we introduce ICL-Bandit, a practical and effective approach that leverages ICL to enhance the query-ad relevance labeling capabilities of LLMs. We develop a novel bandit learning method to identify and provide superior demonstrations for ICL, thereby improving labeling performance. Experimental results demonstrate that ICL-Bandit achieves state-of-the-art performance compared to existing methods. Additionally, ICL-Bandit has been deployed in Company X[1] that serves billions of users worldwide, confirming its robustness and effectiveness.

1 Introduction

In advertisement (ad) recommendation systems such as Microsoft Bing Ads and Google Ads, high-quality labeled data is of critical importance for training ad recommendation models, especially labeling the relevance between user query text and ad description text, as discussed in (Ling et al., 2017; Shuai et al., 2020; Wang et al., 2022a). The traditional approach is human labeling which is costly and inefficient. This is particularly challenging given the huge amount of data to be labelled, and labeling such relevance between user query and ad requires a good knowledge and experience. For example,

> **User Query:**
> *"Innovative treatments for reducing hospital readmission rates in heart failure patients."*
>
> **Advertisement:**
> *"Remote Patient Monitoring Systems - Continuous Care for Heart Failure Patients"*

This query-ad pair is labeled as relevant since remote patient monitoring systems provide continuous care and real-time health data, enabling proactive management of heart failure, which is essential for reducing hospital readmission rates, and such manual labeling requires domain knowledge.

Recent advances in Large Language Models (LLMs) have shown that LLMs are highly aligned with human judgments and even surpass human performance in certain tasks (Ouyang et al., 2022), such as topic identification and twitter relevance for political issues (Gilardi et al., 2023), general question-answering data generation (Meng et al., 2023), and instruction data generation (Wang et al., 2022b). However, the lack of domain knowledge limits the performance of LLM in the query-ad relevance labeling task. To address this challenge, many approaches employed in-context learning (ICL) to incorporate domain-specific knowledge as extra context in the LLM's prompt (Kossen et al., 2023; Dong et al., 2022). Besides, it is well known that the effectiveness of ICL heavily depends on the quality of the provided demonstrations, which has motivated many works to explore effective demonstration retrieval methods for ICL, such as (Rubin et al., 2021; Li et al., 2023; Wu et al.; Zhang et al., 2022), all of these methods aim to retrieve better examples from annotated training sets to enhance LLMs' domain knowledge.

Previous work on demonstration retrieval falls

[1] We use Company X for anonymity review.

into two categories. One category involves off-the-shelf retrievers like BM25 (Robertson et al., 2009) or KNN (Guo et al., 2003), which can retrieve textually or semantically similar demonstrations. The other category focuses on training task-specific retrievers with positive and negative demonstrations. Notable examples include Rubin *et al.* (Rubin et al., 2021), Shi *et al.* (Shi et al., 2022), and Xiaonan *et al.* (Li et al., 2023), who leverage LLM feedback (compare the labels generated by LLM with the ground-truth labels, using them as the training signal) to train these retrievers via supervised or contrastive learning. However, the vast combination space of different demonstrations and queries poses a challenge. Randomly sampling demonstrations to collect the LLM's feedback may lead to large parts of "less useful" examples. Some methods, like Zhang *et al.* (Zhang et al., 2022) and Mingkai *et al.* (Deng et al., 2022), employ reinforcement learning to actively sample demonstrations and obtain LLM feedback. But these methods are limited in considering only a fixed number of candidate demonstrations, which reduces the action space for policy training.

To overcome the challenges addressed above in SOTA methods, at first, we frame demonstration retrieval problem as a multi-armed bandit (MAB) problem (Lai and Robbins, 1985), and bandit algorithms solving MAB problems (Vermorel and Mohri, 2005; Li et al., 2010) have demonstrated excellent performance in addressing exploration and exploitation dilemma when dealing with large-scale search spaces. This allows us to design effective exploration techniques for sampling demonstrations and obtaining LLM feedback during retriever training. Then, we propose a novel in-context learning (ICL) algorithm, called *ICL-Bandit*, which leverages a stochastic bandit algorithm to empower ICL at scale with diverse demonstration pools. The objective of ICL-Bandit is to retrieve demonstrations and maximize cumulative positive LLM feedback over a series of retrievals. Figure 1 shows a comparison on the example query-ad pair with demonstrations retrieved with KNN and our ICL-Bandit, respectively.

Our contributions can be summarized as follows:

- We formulate demonstration retrieval as a multi-armed bandit problem, focusing on effective retrieval during retriever training.
- We design a stochastic bandit algorithm suitable for ICL with a large and varied demonstration pool.
- Our approach achieves SOTA performance comparing to other existing methods and it has been deployed to the labeling process at Company X, resulting in substantial cost savings by automated labeling.

2 ICL-Bandit with large and varied demonstration pool

In this section, we introduce ICL-Bandit, a stochastic bandit algorithm designed to efficiently retrieve demonstrations and collect LLM feedback during retriever training. The task is to precisely label query-ad relevance with ICL, and our goal is to train a demonstration retriever which retrieves good demonstrations for ICL. When training the retriever using LLM feedback (we compare the output labels generated by the LLM with the ground-truth labels, employing them as the reward signal), the key lies in how to effectively retrieve demonstrations from a large and diverse pool during the training process. Addressing the Exploration (searching for diverse and potentially informative demonstrations) versus Exploitation (retrieving high-reward demonstrations) balance is pivotal. To tackle this challenge, we first formulate the demonstration retrieval task as a bandit problem.

2.1 Task Definition

The task is to label the relevance of query-ad pairs leveraging LLMs. Compared with zero-shot LLM labeling, providing with demonstrations as context in the ICL manner improves the labeling performance. The ICL prompt comprises four key components:

- **Instruction** We employ the following instruction to describe labeling requests: "*Given user query and an ad, assign a label based on following definitions: - 'Relevant': The ad content directly addresses the user's query, providing information or a solution that aligns with the search intent. - 'Irrelevant': The ad content does not address the user's query, failing to provide information or a solution that matches the search intent.*" This instruction guides the relevance labeling process for LLMs.
- **Input** The input component specifies the query and ad requiring labeling. For instance, "*User*

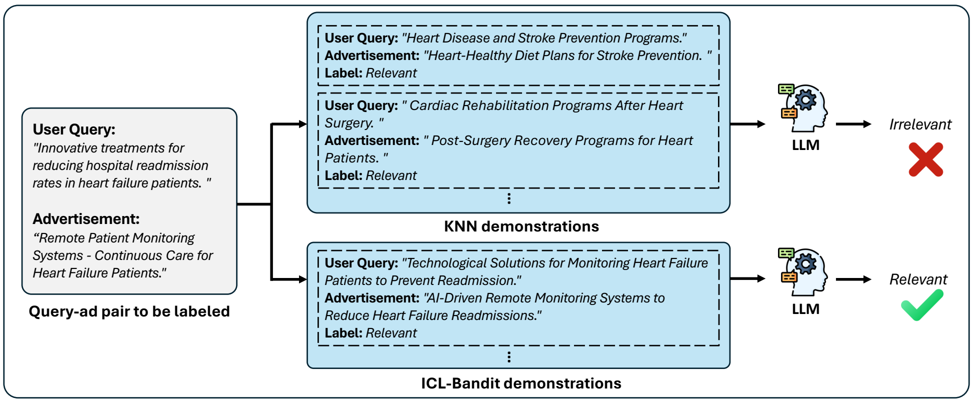

Figure 1: An example illustrating how LLM labeler prompting, when combined with demonstrations retrieved via KNN and our ICL-Bandit approach, yields distinct labeling outcomes.

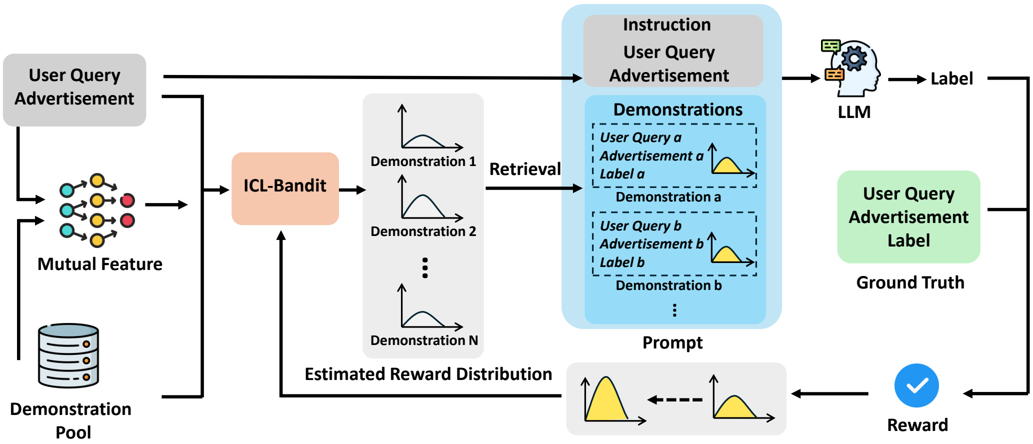

Figure 2: An illustration on training demonstration retrieval process of ICL-Bandit. The demonstration pool consists of expert labeled demonstrations, and each demonstration includes user query, advertisement, and label. For each query-ad from the training set, ICL-Bandit retrieves demonstrations from the demonstration pool considering the estimated reward distribution. With the retrieved demonstrations as the context, LLM labels the query-ad as relevant or irrelevant. Then the LLM-generated label is compared with the ground truth label from the training set to give feedback. In the figure, the positive reward (matched label) is used to update ICL-Bandit to refine the retrieval policy.

Query: 'Student loans suspended until september'; Advertisement:'10 Best Student Loan Refinance'". This information sets the context for the query-ad labeling task.

- **Demonstrations** Demonstrations consist of a set of labeled demonstrations, such as, "*User Query: 'School registration', Advertisement: 'Find Virtual School Programs', Label: Relevant*". Different retrieved demonstrations would highly affect the labeling performance. These examples, provided by human annotators, serve as training instances for a policy π_θ that leverages LLM feedback to retrieve appropriate demonstrations for each unique input.

- **Output Indicator** The output indicator instructs the LLM to generate the labeling answer. For instance, "*Return your decision on the label in <Label></Label> tags.*". This guides the LLM to generate the final label.

2.2 Problem Formulation

The key of the labeling task lies in retrieving informative demonstrations for ICL labeling, then we formulate the demonstration retrieval problem as a multi-armed bandit (MAB) problem, drawing an analogy to the scenario of a gambler selecting from a slot machine with multiple arms in a casino. The player's objective is to choose the arm that offers the highest expected gain. Each time the player pulls an arm and receives a gain or not, they update their estimation of the arm's potential gain. Similar to this scenario, in the query-ad relevance labeling task, we propose a MAB approach to retrieve demonstrations from a pool to achieve a good performance. We define key components such as states, arms, rewards, and the overall objective. **Trial**: In each trial t, our goal is to retrieve m demonstration from the demonstration pool. **State**: State s_t represents the current contextual environment. In the context of ICL, s_t denotes the embedding of the query and ads that require labeling. **Arm/Action**: Each arm in the MAB problem corresponds to a potential demonstration, representing different choice or action that can be taken during the ICL labeling. **Reward**: The reward $r_{(t,a_k)}$ provides numerical feedback, indicating whether the LLM assigns the correct relevance label based on the retrieved demonstration a_k. We define two reward options: a continuous reward $r_{(t,a_k)} \in [0, 1]$ (representing the probability of the LLM's output label) and a discrete reward $r_{(t,a_k)} \in 0, 1$. In the discrete case, a correct label receives a reward of 1, while an incorrect label receives 0. **Objective**: The overall goal is to learn a retrieval policy π_θ by maximizing the cumulative reward over a series of trials during training.

The process of using a MAB algorithm to efficiently retrieve demonstrations during training consists of the following steps:

Step 1: Retrieval of Demonstrations At each trial t, the retrieval policy π_θ retrieves a demonstration (arm) a_k from a demonstration pool. The optimal retrieval will be conducted based on a balance between the benefits (the mean of a_k's rewards) and the chance (the variance of a_k's rewards).

Step 2: Reward from LLM Feedback After retrieving a demonstration a_k, the retriever receives a reward r_t, indicating the correctness of the label provided by the LLM with the chosen demonstration.

Step 3: Policy Update The collected reward value is used to update the policy parameters π_θ to maximize cumulative reward during training.

These steps iteratively occur for a pre-defined number of iterations ($T = 2000$ in our paper). The exploration-exploitation trade-off in Step 1 is crucial, requiring the retrieval policy to balance exploring new demonstrations for potential benefits (exploration) and exploiting known good demonstrations to maximize the mean of rewards (exploitation). This exploration-exploitation balance ensures the effectiveness of learning the demonstration retrieval policy in ICL labeling.

2.3 ICL-Bandit

In this section, we introduce ICL-Bandit as an innovative approach to address the challenges encountered by previous bandit algorithms. Traditional bandit algorithms, which assign a parameter θ to each arm or action, facing the limitations when applied to demonstration retrieval due to the expansive and varied nature of the available demonstrations.

We leverage the framework of Stochastic Multi-Armed Bandit (Bubeck et al., 2012), a variant of the classical MAB problem where rewards associated with different actions (referred to as "arms") are influenced by stochastic processes. In our context, we develop a novel stochastic bandit algorithm tailored to scenarios with an extensive and diverse set of demonstrations.

First, we fine-tune a BERT model in the Company X's query and ads dataset, resulting in an embedding vector e_{a_k} unique to each demonstration a_k, an embedding vector e_{s_t} for the state s_t at trial t and a mutual embedding $e_{(s,a)_t}$. Then we have a unified feature embedding $x_{s_t,a_k} = [e_{s_t}, e_{a_k}, e_{(s,a)_t}]$ to capture contextual information. Next, we adopt a shared parameter θ applicable to all demonstrations. This shared parameterization streamlines the learning process, enhancing efficiency and generalization across the diverse pool of demonstrations.

2.3.1 Demonstration and State Representation Learning

To integrate both the state and demonstration into a unified embedding vector, we employ a self-supervised learning approach to fine-tune a 24-layer BERT model using Company X's user query-ad dataset. The final embedding is derived by extracting the output of the last hidden layer, serving as a comprehensive representation of both the state and demonstration.

2.3.2 Objective Function of ICL-Bandit

Throughout the total T trials, the cumulative reward is defined as $\sum_{t=1}^{T} r_{(t,a_k)}$. In this context, we establish the optimal expected T-iteration reward, denoted as $E[\sum_{t=1}^{T} r_{(t,a_k^*)}]$, where a_k^* represents the optimal demonstration yielding the maximum expected reward at trial t. Our objective is to proficiently retrieve a sequence of demonstrations during training, maximizing the expected total payoff. Alternatively, our aim is to minimize the regret of the algorithm concerning the optimal demonstration retrieval strategy. The T-iteration regret of ICL-Bandit can be formally defined as:

$$Re(T) = E[\sum_{t=1}^{T} r_{(t,a_k^*)}] - E[\sum_{t=1}^{T} r_{(t,a_k)}] \quad (1)$$

2.3.3 Optimization of ICL-Bandit

To minimize the regret, it is assumed that the expected reward of an example a is linear in the d-dimensional state-action integrated feature $x_{(t,a)}$, with three unknown policy parameters θ^*_{state}, θ^*_{action}, and θ^*_{mut}:

$$\mathbb{E}[r_{(t,a)} \mid x_{(t,a)}] = x^T_{(t,a)}[\theta^*_{\text{state}}, \theta^*_{\text{action}}, \theta^*_{\text{mut}}]$$

where θ^*_{state} denotes the policy parameter for the context or the state, i.e., for the target sample, θ^*_{action} denotes the policy parameter for an action, i.e., for an example, and θ^*_{mut} denotes the policy parameter for mutual information of the target sample and the example. The mutual information may be common or similar information between the target sample and the example. The policy parameter θ^*_{state} can map the target sample to a first vector space. The policy parameter θ^*_{action} can map the example to a second vector space. The first vector space and the second vector space are different and independent, but they are dual to each other. The policy parameter θ^*_{mut} can map both the target sample and the example to the same vector space. The technical effect of using the three policy parameters θ^*_{state}, θ^*_{action}, and θ^*_{mut} is to more accurately measure the relationship or distance between the target sample and the example, so as to calculate a more accurate expected reward.

The embodiments of the present disclosure propose that all examples share three policy parameters θ^*_{state}, θ^*_{action}, and θ^*_{mut}. This parameterization remains constant regardless of the number of examples. The technical effect of such settings is to streamline the learning process, and enhance efficiency and generalization across the diverse set of examples. This framework enables the application of the proposed reinforced retrieval operation to large-scale and diverse candidate examples, contributing to its scalability and adaptability.

For each example, we have three kinds of features: state, action, and mut, denoted as $[e_{s_t}, e_{a_k}, e_{(s,a)_t}]$. These correspond to the data matrices D_{state}, D_{action}, and D_{mut}, which represent samples on different features. Let D_{state}, D_{action}, and D_{mut} be data matrices of dimension $m \times d$ at trial t, where the rows correspond to m training inputs of context, action, and mutual information, and $b \in \mathbb{R}^m$ is the corresponding reward vector (e.g., the m rewards indicating whether the LLM provided the correct label in the training set). Applying ridge regression to the training data (D, b) yields an estimate of the policy parameters:

$$\theta_{\text{state}} = \left(D^T_{\text{state}} D_{\text{state}} + \lambda I\right)^{-1} D^T_{\text{state}} b \quad (2)$$

$$\theta_{\text{action}} = \left(D^T_{\text{action}} D_{\text{action}} + \lambda I\right)^{-1} D^T_{\text{action}} b \quad (3)$$

$$\theta_{\text{mut}} = \left(D^T_{\text{mut}} D_{\text{mut}} + \lambda I\right)^{-1} D^T_{\text{mut}} b \quad (4)$$

where I is the $d \times d$ identity matrix, $\lambda \in [0, 1]$ is the regularization term of ridge regression estimation. Let $D = [D_{\text{state}}, D_{\text{action}}, D_{\text{mut}}]$, and $\theta = [\theta_{\text{state}}, \theta_{\text{action}}, \theta_{\text{mut}}]$. When components in b are independent conditioned on corresponding rows in D, it can be shown that, with probability at least $1 - \delta$: $\left| x^T_{(t,a_t)} \hat{\theta} - \mathbb{E}[r_{(t,a_t)} \mid x_{(t,a_t)}] \right| \leq \alpha' \sqrt{x^T_{(t,a_t)} \left(D^T D + \lambda I\right)^{-1} x_{(t,a_t)}}$ For any $\delta > 0$ and $x_{(t,a_t)} \in \mathbb{R}^d$, where $\hat{\theta}$ is the mean of θ, and a_t indicates the example selected at t, $r_{(t,a)}$ is the observed reward, σ^2 is the variance proxy of the noise and α' is a constant. Details of the proof is provided in Appendix A.

2.4 ICL-Bandit vs. Traditional Bandit

ICL-Bandit improves upon traditional bandit methods by introducing shared parameters, θ^*_{state}, θ^*_{action}, and θ^*_{mut}, to jointly model state, actions, and their interactions. This enables better alignment between context and candidate demonstrations, leading to more accurate action selection.

Unlike traditional methods that treat actions independently, ICL-Bandit captures complex contextual dependencies, enhances generalization, and scales efficiently to high-dimensional data. Its unified framework ensures consistent performance across diverse ICL labeling tasks, mitigating the inconsistency and overfitting often seen in conventional approaches.

3 Experiment

Dataset: We use a high-quality, expert-labeled dataset collected daily over 1.5 years, consisting of user queries and associated advertisement information (e.g., keywords, titles, descriptions, URLs), each labeled as relevant or irrelevant. The dataset is temporally split into: Example pool: 1,578,728 samples used as demonstrations. Training set: 9,999 query-ad pairs. Test set: 1,986 query-ad pairs. This temporal partitioning simulates real-world deployment, where models are trained on historical data and evaluated on recent, unseen examples.

Evaluation Metrics: We assess binary classification performance using Accuracy (ACC), F1-score, Precision, and Recall to capture both correctness and balance in predictions. More details of the experimental settings are provided in Appendix C.

3.1 Competitors

For a fair comparison, all baselines and ICL-Bandit (except "No Example" and "Crowdsourcing") were provided with 3 positive (Relevant) and 3 negative (Irrelevant) historically labeled samples as demonstrations. The following methods were selected as our competitors: **No Example (Zero-shot Learning)**, **Crowdsourcing**, **EPR (SL-KNN)**, **EPR (SL-LLM) (Li et al., 2023)**, **Q-learning (Zhang et al., 2022)**, **Static**, **BM25** (Robertson et al., 2009), **Random**, **KNN** (Guo et al., 2003). Details of the baseline methods are provided in Appendix B.

Table 1: Results of GPT-3.5 as the backbone LLM.

Model	ACC (%)	F1-score (%)	Precision (%)	Recall (%)
No Example	53.95	25.72	41.76	18.58
Crowdsourcing	**67.57**	**73.65**	**90.09**	**62.28**
EPR (SL-KNN)	55.97	25.68	39.22	19.09
EPR (SL-LLM)	57.18	25.44	41.55	18.33
Q-learning	58.68	34.71	43.78	28.14
Static	56.57	28.56	43.40	21.28
BM25	58.84	28.77	46.35	20.86
Random	50.73	36.49	37.52	35.52
KNN	57.98	27.60	44.04	20.10
ICL-Bandit (Ours)	63.76	61.63	65.38	58.29

Table 2: Results of GPT-4 as the backbone LLM.

Model	ACC (%)	F1-score (%)	Precision (%)	Recall (%)
No Example	65.05	61.45	64.08	59.03
Crowdsourcing	67.57	73.65	**90.09**	62.28
EPR (SL-KNN)	74.42	80.70	73.49	89.47
EPR (SL-LLM)	74.62	80.70	72.94	**90.32**
Q-learning	74.26	76.25	78.81	87.80
Static	73.56	79.71	71.35	90.28
BM25	73.62	80.09	72.94	88.80
Random	73.72	79.94	71.97	89.89
KNN	74.47	80.74	73.56	89.48
ICL-Bandit (Ours)	**80.03**	**82.57**	76.91	89.14

3.2 Results Analysis

Tables 1 and 2 present the experimental results comparing nine demonstration retrieval methods, including our ICL-Bandit, across two versions of LLMs. The analysis highlights key performance trends. The "No Example" baseline performs poorly, while "Crowdsourcing" demonstrations achieve the highest accuracy and precision, emphasizing the importance of expert-labeled data. Among automated methods, Q-learning, EPR (SL-LLM), and ICL-Bandit show strong performance, benefiting from LLM feedback. Notably, ICL-Bandit surpasses Q-learning and EPR (SL-LLM) despite using only 2,000 feedback samples compared to their 5,000, due to its lightweight, linear design that requires fewer data.

EPR (SL-KNN) and EPR (SL-LLM) improve over the "No Example" baseline but still lag behind "Crowdsourcing," indicating that retrieval effectiveness depends on technique selection. Similarly, methods like "Static," "Random," "KNN," and "BM25" show varied performance, with BM25 performing competitively but still unable to match expert-labeled demonstrations.

ICL-Bandit consistently delivers superior results, often outperforming or matching "Crowdsourcing." Its ability to balance exploration and exploitation allows it to retrieve relevant demonstrations effectively, adapt to diverse queries, and enhance recall, improving overall ICL performance.

3.3 Learning Curve of ICL-Bandit

The learning curve experiment was devised to examine the evolutionary performance of ICL-Bandit as training data accumulates. The primary objective was to discern how the method's effectiveness scales with an expanding dataset, providing insights into its adaptability and scalability. The experiment's results are depicted in Figure 3, where the x-axis represents training iterations, and the y-axis portrays the cumulative mean and variance of Accuracy, Binary Accuracy, True Negative Rate (TNR), and True Positive Rate (TPR). The learning curve analysis of ICL-Bandit highlights its capacity to dynamically adapt and enhance its performance over successive training iterations. Notably, it illustrates that ICL-Bandit achieves a rapid and stable convergence to a commendable performance level.

Furthermore, the outcomes suggest that ICL-Bandit exhibits promise for demonstration retrieval in ICL, even when trained on a limited LLM feedback dataset. Remarkably, in comparison to EPR (SL-LLM), which utilized a larger dataset of 5000 feedback instances, ICL-Bandit demonstrates superior performance. The learning curve analysis underscores the efficacy of ICL-Bandit in iteratively improving its performance with an increasing volume of training data. This positions it as a robust and scalable solution for the nuanced task

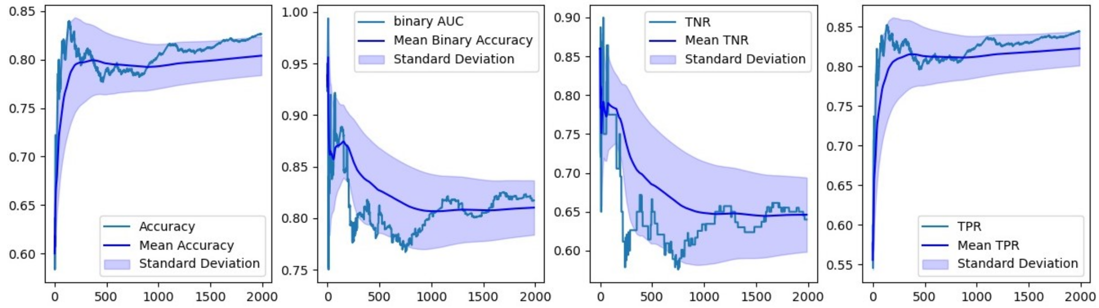

Figure 3: The learning curve of ICL-Bandit during 2000 trails training. TNR and TPR indicates the true negative rate and true positive rate respectively.

of demonstration retrieval in complex information retrieval scenarios.

3.4 Ablation Study

3.4.1 Number of retrivaled samples

Example number	ACC	Binary AUC	TNR	TPR
1	0.7826	0.8127	0.8326	0.6978
3	0.8003	0.8029	0.8215	0.7077
6	0.8001	0.8127	0.8178	0.7129
9	0.7697	0.7616	0.7716	0.7516

Table 3: Performance metrics on different number of selected demonstrations.

In this experiment, we evaluate the impact of varying the number of positive and negative demonstrations on model performance. The results, presented in Table 3, indicate that the performance metrics (ACC, Binary AUC, TNR, and TPR) generally improve as the number of positive/negative demonstrations increases from 1 to 3. Specifically, the best overall performance is observed when 3 demonstrations are used. Thus we choose 3 as the final number.

When 9 demonstrations are used, the performance metrics begin to decline, indicating that adding too many demonstrations may lead to diminishing returns or even reduced performance.

3.4.2 Training Epochs and Reward Types

The experiment evaluates the performance of the ICL-bandit approach under two reward settings: continuous and discrete. An epoch is defined as a complete pass through the training data. During each epoch, the ICL-bandit retrieves informative demonstrations, selects the best actions, and updates its retrieval policy based on the rewards received. The results in Figure 4 illustrate how the number of epochs affects performance across various metrics.

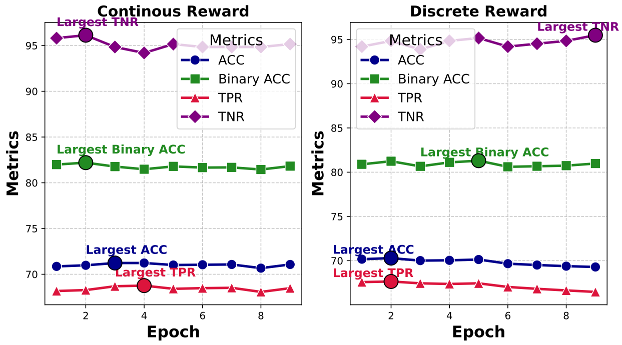

Figure 4: Performance on different epochs and reward types.

We observe that ICL-bandit's performance varies with the number of epochs, with different metrics reaching their optimal levels at different stages. The continuous reward setting, which provides more detailed feedback, achieves peak performance in fewer epochs compared to the discrete reward setting. This suggests that using continuous rewards in practice can reduce training complexity while still delivering strong performance. Finally, we choose continous reward with 1 epoch for reducing the complexity and promising results.

3.5 Application in Practice

We deployed the ICL-Bandit approach in Company X's ad relevance pipeline to reduce manual labeling costs and enhance ad recommendation quality. Each day, the system collects fresh user queries and ads, cleans them using Bing's distributed platform, and applies ICL-Bandit for automated labeling. For each query-ad pair, we retrieve 3 relevant and 3 irrelevant historical examples to construct prompts, which are then labeled using GPT-4. This process generates high-quality labeled data daily for downstream CTR prediction (Lee et al., 2023).

As shown in Tables 4and 5, ICL-Bandit consistently outperforms baselines in both English and non-English settings, demonstrating robust im-

Table 4: A/B testing on English datasets with GPT-4 as the backbone LLM.

Model	ACC (%)	F1-score (%)	Precision (%)	Recall (%)
No Example	66.82	62.18	63.74	60.75
KNN	76.52	75.12	78.72	71.81
ICL-Bandit (Ours)	**87.12**	**82.28**	**86.95**	**78.17**

Table 5: A/B testing on non-English dataset with GPT-4 as the backbone LLM.

Model	ACC (%)	F1-score (%)	Precision (%)	Recall (%)
No Example	63.26	59.48	60.17	58.79
KNN	70.67	75.44	70.86	80.81
ICL-Bandit (Ours)	**80.67**	**85.68**	**82.57**	**89.14**

provements in accuracy, F1-score, precision, and recall.

3.6 Impact on Ad Recommendation

Integrating ICL-Bandit-labeled data into Bing's CTR prediction model led to significant business gains. Offline evaluation on 500K historical query-ad pairs showed a 2.5% AUC increase and 1.8% reduction in Log Loss. In two weeks of online A/B testing with 2 million users, CTR rose by 3.2% and conversion rates improved by 2.7%. Beyond performance, the automated labeling process reduced manual annotation costs by 61%, enabling scalable and cost-effective data processing across millions of queries daily. The results of the A/B testing on English and Non-English datasets are summarized in Table 4 and Table 5.

4 Related Work

4.1 LLM Labeling

Latest studies in LLM have shown that LLM is highly consistent with human judgments and even outperforms humans in many tasks, for example, topic identification and twitter relevance for political issues (Gilardi et al., 2023), general question-answering data generation (Meng et al., 2023), instruction data generation (Wang et al., 2022b) and RL from AI feedback (RLAIF) (Lee et al., 2023). A set of work using LLM for labeling instead of human (Tan et al., 2024; Alaofi et al., 2024; Artemova et al., 2024). In this work, we focus on a domain-specific labeling problem, *i.e.*, query-ad relevance labeling, which requires domain knowledge to guide LLM for labeling.

4.2 Demonstration Retrieval for In-Context Learning

LLMs have emerged as a pivotal strategy for addressing tasks specific to particular domains. However, the effectiveness of ICL is intrinsically tied to the quality of the provided demonstrations (Li et al., 2023; Wu et al.; Zhang et al., 2022). Works such as (Rubin et al., 2021; Li et al., 2023; Wu et al.; Zhang et al., 2022) collectively aim to optimize the retrieval of exemplary instances from annotated training sets, thereby enhancing the domain knowledge encapsulated by LLMs.

Existing demonstration retrieval methods are typically categorized into utilization of off-the-shelf retrievers such as BM25 (Robertson et al., 2009) or KNN (Guo et al., 2003), or training task-specific retrievers using positive and negative demonstrations (Rubin et al., 2021; Shi et al., 2022; Li et al., 2023). These researchers leverage LLM feedback to guide the training of these retrievers through supervised or contrastive learning. Despite these advancements, the vast combinatorial space encompassing different demonstrations and queries presents a significant challenge. Randomly sampling demonstrations to collect LLM feedback risks incorporating a substantial portion of less useful examples. Reinforcement learning-based methods (Zhang et al., 2022; Deng et al., 2022) actively sample demonstrations and elicit valuable LLM feedback. However, they are constrained by a fixed number of demonstrations, thereby limiting the action space available for policy training.

5 Conclusion

In this paper, we leverage LLMs to automate query-ad relevance labeling for improved ad recommendation. To address the lack of domain-specific knowledge in LLMs, we adopt in-context learning (ICL) and propose ICL-Bandit, a stochastic bandit algorithm for retrieving high-quality demonstrations and collecting LLM feedback to train a retriever. Our approach outperforms existing retrieval methods and has been successfully deployed in Company X's ad recommendation system, delivering significant cost savings and strong real-world effectiveness.

6 Limitations

ICL-Bandit's performance heavily relies on the quality and coverage of the labeled demonstration pool. If the pool lacks diverse or representative examples for certain query-ad pairs, the retrieved demonstrations may be suboptimal, limiting the effectiveness of in-context learning. This constraint can affect generalization, especially in long-tail or evolving domains where labeled data is sparse or outdated.

References

Marwah Alaofi, Paul Thomas, Falk Scholer, and Mark Sanderson. 2024. Llms can be fooled into labelling a document as relevant: best café near me; this paper is perfectly relevant. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 32–41.

Ekaterina Artemova, Akim Tsvigun, Dominik Schlechtweg, Natalia Fedorova, Sergei Tilga, and Boris Obmoroshev. 2024. Hands-on tutorial: Labeling with llm and human-in-the-loop. *arXiv preprint arXiv:2411.04637*.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and 1 others. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P Xing, and Zhiting Hu. 2022. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. 2003. Knn model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pages 986–996. Springer.

Jannik Kossen, Tom Rainforth, and Yarin Gal. 2023. In-context learning in large language models learns label relationships but is not conventional learning. *arXiv preprint arXiv:2307.12375*.

Tze Leung Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670.

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*.

Xiaoliang Ling, Weiwei Deng, Chen Gu, Hucheng Zhou, Cui Li, and Feng Sun. 2017. Model ensemble for click prediction in bing search ads. In *Proceedings of the 26th international conference on world wide web companion*, pages 689–698.

Yu Meng, Martin Michalski, Jiaxin Huang, Yu Zhang, Tarek Abdelzaher, and Jiawei Han. 2023. Tuning language models as training data generators for augmentation-enhanced few-shot learning. In *International Conference on Machine Learning*, pages 24457–24477. PMLR.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. 2022. Xricl: Cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-sql semantic parsing. *arXiv preprint arXiv:2210.13693*.

Jiamei Shuai, Shuayb Zarar, and Denis Charles. 2020. Modeling and evaluation framework for responsive search ads. In *Machine Learning, AI & Data Science Conference (MLADS)*. Microsoft.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957.

Joannes Vermorel and Mehryar Mohri. 2005. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*, pages 437–448. Springer.

Dong Wang, Shaoguang Yan, Yunqing Xia, Kavé Salamatian, Weiwei Deng, and Qi Zhang. 2022a. Learning supplementary nlp features for ctr prediction in sponsored search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4010–4020.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022b. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

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

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

A Proof

A.1 Step 1: Decompose the Estimation Error

The estimation error can be expressed as:

$$\begin{aligned}\hat{\theta}-\theta^* &= \left(D^\top D+\lambda I\right)^{-1} D^\top b-\theta^* \\ &= \left(D^\top D+\lambda I\right)^{-1} D^\top\left(D\theta^*+\epsilon\right)-\theta^* \\ &= \left(D^\top D+\lambda I\right)^{-1} D^\top D\theta^*+\left(D^\top D+\lambda I\right)^{-1} D^\top\epsilon-\theta^* \\ &= \left[\left(D^\top D+\lambda I\right)^{-1} D^\top D-I\right]\theta^*+\left(D^\top D+\lambda I\right)^{-1} D^\top\epsilon.\end{aligned}$$

Simplifying:

$$\hat{\theta}-\theta^*=-\lambda\left(D^\top D+\lambda I\right)^{-1}\theta^*+\left(D^\top D+\lambda I\right)^{-1} D^\top\epsilon.$$

A.2 Step 2: Express the Estimation Error Components

Let: **Bias Term:** $\text{Bias}=-\lambda\left(D^\top D+\lambda I\right)^{-1}\theta^*$.

Variance Term: $\text{Variance}=\left(D^\top D+\lambda I\right)^{-1} D^\top\epsilon$.

Then: $\hat{\theta}-\theta^*=\text{Bias}+\text{Variance}$.

A.3 Step 3: Bound the Bias Term

We aim to bound $\left|x_{(t,a)}^\top\text{Bias}\right|$.

Using the Cauchy-Schwarz inequality:

$$\left|x_{(t,a)}^\top\text{Bias}\right|=\lambda\left|x_{(t,a)}^\top\left(D^\top D+\lambda I\right)^{-1}\theta^*\right| \quad (5)$$

$$\leq\lambda\left\|x_{(t,a)}\right\|_{\left(D^\top D+\lambda I\right)^{-1}}\left\|\theta^*\right\|, \quad (6)$$

where $\|x\|_A=\sqrt{x^\top Ax}$ denotes the Mahalanobis norm with respect to the matrix A.

Assuming $\|\theta^*\|\leq S$, where S is a known bound on the norm of θ^*, we have:

$$\left|x_{(t,a)}^\top\text{Bias}\right|\leq\lambda S\left\|x_{(t,a)}\right\|_{\left(D^\top D+\lambda I\right)^{-1}}. \quad (7)$$

A.4 Step 4: Bound the Variance Term

We aim to bound $\left|x_{(t,a)}^\top\text{Variance}\right|$ with high probability.

Since ϵ has independent components with zero mean and variance proxy σ^2, the variance of $x_{(t,a)}^\top\text{Variance}$ is:

$$\begin{aligned}\text{Var}\left(x_{(t,a)}^\top\text{Variance}\right) &= \text{Var}\left(x_{(t,a)}^\top\left(D^\top D+\lambda I\right)^{-1} D^\top\epsilon\right) \\ &= \sigma^2 x_{(t,a)}^\top\left(D^\top D+\lambda I\right)^{-1} D^\top D\left(D^\top D+\lambda I\right)^{-1} x_{(t,a)}.\end{aligned}$$

Therefore:

$$\text{Var}\left(x_{(t,a)}^{\top}\text{Variance}\right) = \sigma^2 x_{(t,a)}^{\top}\left(\left(D^{\top}D+\lambda I\right)^{-1}\left(I-\lambda\left(D^{\top}D+\lambda I\right)^{-1}\right)\right)x_{(t,a)} \leq \sigma^2 \left\|x_{(t,a)}\right\|^2_{(D^{\top}D+\lambda I)^{-1}}.$$

A.5 Step 5: Apply Concentration Inequality

Since $x_{(t,a)}^{\top}$Variance is a linear combination of independent sub-Gaussian variables, it is sub-Gaussian with parameter $\sigma' = \sigma \left\|x_{(t,a)}\right\|_{(D^{\top}D+\lambda I)^{-1}}$.

Using a sub-Gaussian tail bound, for any $\delta > 0$:

$$P\left(\left|x_{(t,a)}^{\top}\text{Variance}\right| \geq \alpha \left\|x_{(t,a)}\right\|_{(D^{\top}D+\lambda I)^{-1}}\right) \leq \delta, \quad (8)$$

where $\alpha = \sigma\sqrt{2\ln\left(\frac{1}{\delta}\right)}$.

Step 6: Combine Bias and Variance Terms

The total estimation error is:

$$\left|x_{(t,a)}^{\top}\left(\hat{\theta}-\theta^*\right)\right| \leq \left|x_{(t,a)}^{\top}\text{Bias}\right| + \left|x_{(t,a)}^{\top}\text{Variance}\right|. \quad (9)$$

A.6 Step 7: Final Inequality

Combine the bounds:

$$\left|x_{(t,a)}^{\top}\left(\hat{\theta}-\theta^*\right)\right| \leq (\lambda S+\alpha)\left\|x_{(t,a)}\right\|_{(D^{\top}D+\lambda I)^{-1}}. \quad (10)$$

For sufficiently small λ and bounded θ^*, the bias term can be controlled, and the dominant term becomes the variance term.

Therefore, we can simplify the inequality to:

$$\left|x_{(t,a)}^{\top}\left(\hat{\theta}-\theta^*\right)\right| \leq \alpha' \left\|x_{(t,a)}\right\|_{(D^{\top}D+\lambda I)^{-1}}, \quad (11)$$

where $\alpha' = \lambda S + \alpha$.

B Competitors

For a fair comparison, all baselines and ICL-Bandit (except "No Example" and "Crowdsourcing") were provided with 3 positive (Relevant) and 3 negative (Irrelevant) historically labeled samples as demonstrations. The following methods were selected as our competitors:

- **No Example (Zero-shot Learning)**: Zero-shot learning without any demonstrations.
- **Crowdsourcing**: Demonstrations annotated by human evaluators through crowdsourcing to assess query-ad relevance. It is different from the human (expert) labeled data for demonstraion pool, train and test data.
- **EPR (SL-KNN)**: Demonstrations are retrieved using the K-nearest neighbor (KNN) algorithm based on the training datasets as ground truth. EPR (SL-KNN) is then trained to input query-ads and output the retrieved demonstrations to assist the LLM in labeling.
- **EPR (SL-LLM) (Li et al., 2023)**: Demonstrations are retrieved using GPT-3.5 based on the training datasets as ground truth. EPR (SL-LLM) is then trained to input query-ads and output the retrieved demonstrations to assist the LLM in labeling.
- **Q-learning (Zhang et al., 2022)**: A demonstration candidate is predefined, and Q-learning is utilized to learn the retrieval policy. Demonstrations are clustered into 50 clusters to implement this algorithm.
- **Static**: Demonstrations are pre-defined and kept static.
- **BM25** (Robertson et al., 2009): Demonstrations retrieved using the BM25 algorithm.
- **Random**: Demonstrations randomly sampled for each user query.
- **KNN** (Guo et al., 2003): Demonstrations are retrieved using the K-nearest neighbor (KNN) algorithm based on the user query. We use the same feature embedding as our method to retrieve the demonstrations with cosine similarity in KNN.

C Experimental Setup

Dataset: In the experiments, we leveraged a meticulously curated dataset tailored specifically for assessing the efficacy of demonstration retrieval systems. This dataset is derived from high-quality human (expert)-labeled data collected daily over the recent 1.5-year period. Each sample in the dataset consists of a user query along with associated information about recommended advertisements. This information includes query keywords, ad titles, ad descriptions, ad URLs, and other pertinent content, each labeled as either relevant or irrelevant.

To facilitate a robust evaluation, we partitioned the dataset temporally into three distinct subsets: an example pool, a training set, and a test set. The example pool contains all 1,578,728 samples as demonstrations, ensuring a comprehensive range of instances. For the purpose of training the model, we selected a subset of 9,999 samples specifically for query-ads pair labeling. The evaluation phase was carried out on a test set, which included 1,986 samples also designated for query-ads pair labeling. This temporal division helps in mimicking real-world scenarios where models are trained on historical data and tested on recent, unseen data, thereby providing insights into the practical applicability and performance of the retrieval methods under study.

Evaluation Metric: Our in-context learning method aims to enhance the labeling performance of large language models (LLMs). Given that the labeling task at hand is a binary classification problem, we evaluate the effectiveness of our approach using several key metrics. Specifically, we measure Accuracy (ACC), F1-Score, Precision, and Recall. These metrics collectively provide a comprehensive assessment of the model's performance in terms of both its ability to correctly label data and its balance between precision and recall.

Computational Resource All experiments are performed on single Ubuntu 20.04 LTS system with Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz CPU, 112 Gigabyte memory and single NVIDIA Tesla P100 accelerator.