RETRIEVE LABEL KNOWLEDGE, THEN BOUNDARY RULES: LABEL-CENTRIC RAG FOR TEXT CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Flagship large language models (LLMs) offer rich general knowledge and strong reasoning capabilities, but their high inference cost and latency limit their use for large-scale data labeling. We study zero-shot text classification with semantically meaningful labels and task descriptions, without task-specific labeled examples or classifier feedback. We propose **SLKR** (Semantic Label Knowledge Retrieval), a label-centric retrieval-augmented generation framework that separates offline knowledge construction from lightweight online inference. Offline, a flagship LLM independently generates label definitions, retrieval keywords, and boundary rules for potentially confusable label pairs, forming a reusable label-knowledge store. Because this construction operates over the label space rather than the input corpus, its one-time flagship cost can be amortized across large-scale inference. Online, hybrid semantic and lexical retrieval first identifies the top-K candidate labels and then selects up to M input-relevant boundary rules connecting candidate-label pairs. A lightweight LLM performs the final classification in a single call using the retrieved definitions and rules. Across three public benchmarks and a real-world multilingual book-classification dataset, SLKR achieves the best results on all four datasets with Qwen3.5-4B and on three of four with Ministral-3-3B, while using compact input contexts and no task-specific labeled examples. Boundary ablations further show that retrieving a small set of input-relevant rules preserves performance close to using all boundaries with substantially less context, supporting efficient reuse of flagship-generated label knowledge for lightweight classification.

1 INTRODUCTION

Large language models (LLMs), with their broad knowledge and strong reasoning capabilities, are increasingly used for text annotation and classification. For specialized and broad-coverage data, capable LLMs can outperform non-expert annotators who lack sufficient domain knowledge (Gilardi et al., 2023; Bermejo et al., 2025). However, stronger models generally come with substantially higher inference costs: closed-source models require expensive API calls and may raise data-privacy concerns, while open-source models demand substantial deployment resources and can be slow at inference. These limitations make repeated use of flagship LLMs impractical for large-scale annotation, particularly in industrial cold-start settings where new classification tasks arise frequently, the scale and diversity of the data make comprehensive labeled coverage difficult, and the conventional pipeline of annotation, human verification, and model fine-tuning introduces substantial iteration overhead.

Importantly, many classification labels already carry meaningful semantics. Labels such as *card payment fee charged* in intent classification, or *Environmental Science and Engineering* and *Civil Engineering* in document classification, are not arbitrary identifiers: their meanings provide a natural interface to the general knowledge encoded in pretrained LLMs. This suggests an opportunity to use a capable LLM to elaborate label semantics once, rather than repeatedly relying on it to classify every input. While prior work has shown that label descriptions can improve classification (Kumar et al., 2024), we ask a different question: *Can the knowledge of a flagship LLM be organized into reusable label knowledge, enabling cost-efficient classification with a lightweight LLM?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

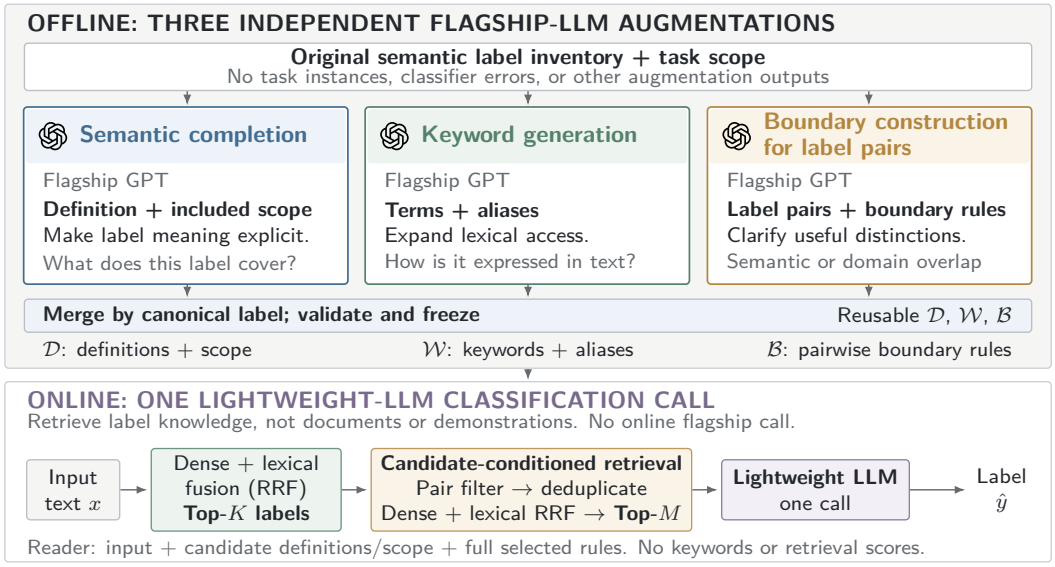

Figure 1: **SLKR overview.** Three independent flagship augmentations receive the same original label inventory and task scope, producing definitions ($\mathcal{D}$), keywords ($\mathcal{W}$), and boundary rules ($\mathcal{B}$). Online, hybrid Top-K label retrieval is followed by candidate-pair filtering, deduplication, and hybrid Top-M rule retrieval. One lightweight LLM reads the input, candidate definitions/scope, and selected rules.

We introduce SLKR, a label-centric RAG formulation for zero-shot text classification. Conventional corpus-grounded RAG retrieves source passages as evidence (Lewis et al., 2020); our input text instead retrieves candidate classes represented by LLM-augmented label knowledge constructed without task-specific labeled examples. The retrieved content consists of class-level decision guidance rather than task documents or labeled demonstrations, shifting retrieval from corpus evidence to reusable knowledge associated with the label space (Table 1).

Figure 1 gives a high-level overview of SLKR. Offline, a flagship LLM independently augments the label space by enriching label definitions, generating retrieval keywords, and constructing boundary rules for potentially confusing labels. These complementary resources capture label semantics, improve candidate retrieval, and help distinguish ambiguous classes. Online, hybrid retrieval first selects a small set of candidate labels and then retrieves the most relevant boundary rules. A lightweight LLM performs the final classification using only the retrieved label knowledge and the input text. Because expensive flagship-model inference is performed only during label-knowledge construction rather than for each input instance, its cost depends on the label space rather than the input corpus and can be amortized over large-scale inference.

Our evaluation asks whether this division of labor achieves competitive classification quality with shorter contexts and lower recurring costs while accounting for offline construction. We evaluate SLKR on public benchmarks and further validate it in a real-world industrial setting involving large-scale document classification. Our main contributions are summarized as follows.

Contributions. Our contributions are threefold:

1. **A label-centric RAG formulation for zero-shot classification.** We introduce a retrieval paradigm that retrieves reusable knowledge about candidate labels rather than source documents or labeled task examples. This shifts retrieval from instance-level evidence or demonstrations to class-level decision knowledge, enabling classification without task-specific labeled data.
2. **Retrieval of explicit decision boundaries between confusable labels.** We use a capable flagship LLM to identify potentially confusing classes and construct boundary rules that capture distinctions beyond label-name similarity, including semantic, domain, functional,

and application-level overlap. At inference time, SLKR retrieves only the boundary knowledge relevant to the input and candidate labels, providing targeted guidance for fine-grained classification.

3. **Scalable lightweight inference through reusable flagship knowledge.** We externalize expensive label understanding to offline knowledge construction and reuse it across inputs, leaving online inference to retrieval and a lightweight LLM. This design substantially reduces recurring inference cost and context requirements while maintaining competitive classification quality, making SLKR suitable for large-scale deployment.

2 Related Work

Label semantics for classification. Prior work has exploited label names and descriptions for text classification. LOTClass and X-Class derive supervision from label names together with unlabeled task corpora (Meng et al., 2020; Wang et al., 2021), while label-description methods enrich class semantics for zero-shot classification (Gao et al., 2023; Kumar et al., 2024). Among these, we compare directly with Gen-Z (Kumar et al., 2024), which is compatible with our instance-free setting. SLKR further organizes definitions, retrieval keywords, and pairwise boundaries into a reusable label-side knowledge base.

Retrieval-augmented classification. Conventional RAG retrieves external documents (Lewis et al., 2020), while retrieved in-context learning selects labeled demonstrations relevant to an input (Liu et al., 2022; Rubin et al., 2022). We include retrieved ICL as a supervised retrieval baseline. Other classification-oriented retrieval methods, such as GORAG (Wang et al., 2026), operate in dynamic few-shot settings with labeled data and therefore assume additional task-instance supervision absent from SLKR.

Boundary-aware classification. Prior work has used pairwise reasoning to distinguish ambiguous labels (Lu et al., 2024). Confusion-aware retrieval further identifies difficult label pairs from classifier errors and generates targeted decision rules (Gupta et al., 2026). These approaches motivate explicit modeling of label boundaries but rely on task instances or classifier feedback. In contrast, SLKR constructs boundary knowledge from the label inventory and task description alone and retrieves input-relevant rules only among the candidate labels.

We next describe how SLKR constructs and retrieves this label-side knowledge for lightweight classification.

3 Label-Centric Retrieval-Augmented Classification

We first define the classification setting and the label-side retrieval object. We then describe SLKR in two phases: offline, a flagship LLM independently constructs semantic definitions, retrieval keywords, and boundary rules; online, hybrid label retrieval selects candidate classes, followed by candidate-conditioned boundary retrieval to identify the most relevant decision rules. A lightweight LLM finally predicts from the retrieved label knowledge. Algorithm 1 summarizes the complete pipeline. We conclude with an analysis of how the one-time flagship construction cost is amortized over large-scale inference.

3.1 Setting and Retrieval Object

Let x denote an input text and $\mathcal{Y} = \{y_1, \ldots, y_N\}$ a closed-set inventory of semantically meaningful labels. The task description and any available taxonomy metadata are denoted by t. Given $(x, \mathcal{Y}, t)$, our goal is to predict a primary label $\hat{y} \in \mathcal{Y}$. Task-specific labeled instances may be used for evaluation and competing few-shot and retrieved in-context learning (ICL) baselines, but are strictly isolated from SLKR's knowledge construction and classifier adaptation.

SLKR uses a frozen flagship LLM F for offline knowledge construction and a frozen lightweight (smaller or lower-cost) LLM g for online classification. For each label y_i, F constructs a label record

$$k_i = (y_i, d_i, w_i, \mathcal{B}_i),$$

Table 1: Comparison of prompting and retrieval-based paradigms for LLM classification. "No External Data" denotes no reliance on labeled examples or an external corpus.

Family	Design	No External Data	Label Knowledge	Boundary Aware	Provided / Retrieved Content
Prompting	Zero-shot	✓	✗	✗	Label names and task description
	Few-shot	✗	✗	✗	Text–label demonstrations
Retrieval-based	Retrieved ICL	✗	✗	✗	Retrieved text–label demonstrations
	Document RAG	✗	✗	✗	Retrieved document passages
	SLKR (ours)	✓	✓	✓	Label knowledge and boundary rules

where d_i is an augmented semantic definition, w_i is a set of retrieval keywords, and $\mathcal{B}_i$ contains boundary rules associated with y_i. Specifically, a boundary rule b_{ij} describes the decision boundary between a potentially confusable pair (y_i, y_j). The resulting knowledge base

$$\mathcal{K} = \{k_i\}_{i=1}^{N}$$

therefore indexes reusable label-side knowledge rather than instances from the corpus being classified.

3.2 Three Independent Flagship Augmentations

Given the original label inventory $\mathcal{Y}$ and task description t, the flagship LLM F independently constructs three complementary types of label knowledge:

$$\mathcal{D} = F_{\text{sem}}(\mathcal{Y}, t), \qquad \mathcal{W} = F_{\text{lex}}(\mathcal{Y}, t), \qquad \mathcal{B} = F_{\text{bnd}}(\mathcal{Y}, t), \tag{1}$$

where the subscripts denote different prompting roles of the same flagship model. Each augmentation is generated directly from $(\mathcal{Y}, t)$ without using the outputs of the other branches or task-specific labeled examples.

Semantic definitions. F_{sem} expands each label y_i into a concise definition d_i describing its semantic scope and representative topics. For example, for a WoS label such as *Computer Science*, the generated definition may describe the study of computation, algorithms, software, computer systems, artificial intelligence, and related computational methods. This provides richer semantic information than the label name alone.

Retrieval keywords. F_{lex} generates a set of label-specific terms w_i to improve lexical retrieval. For *Computer Science*, these may include *algorithm*, *software*, *machine learning*, *database*, and *computer vision*. The keywords are used only for candidate retrieval and are omitted from the final classification prompt to reduce token usage and avoid redundant context.

Boundary rules. Definitions and keywords describe individual labels but do not explicitly resolve ambiguity between related classes. F_{bnd} therefore identifies potentially confusable label pairs and generates natural-language rules b_{ij} describing how they should be distinguished. For example, a boundary between *Computer Science* and *Electrical Engineering* may distinguish work centered on algorithms, software, and computation from work centered on circuits, electronic devices, and electrical systems. Such boundaries can capture overlap in semantics, domains, methods, and applications that is not apparent from label names alone.

The three outputs are combined into the label-side knowledge base $\mathcal{K}$. Once constructed, $\mathcal{K}$ is frozen and reused across all input instances, allowing the cost of flagship-model augmentation to be amortized over subsequent classification.

3.3 Hybrid Label Retrieval

We next describe how the augmented label knowledge is used to retrieve candidate labels. SLKR employs hybrid retrieval that combines semantic and lexical signals. For semantic retrieval, a lightweight frozen encoder h maps the input x and each augmented label definition d_i into the

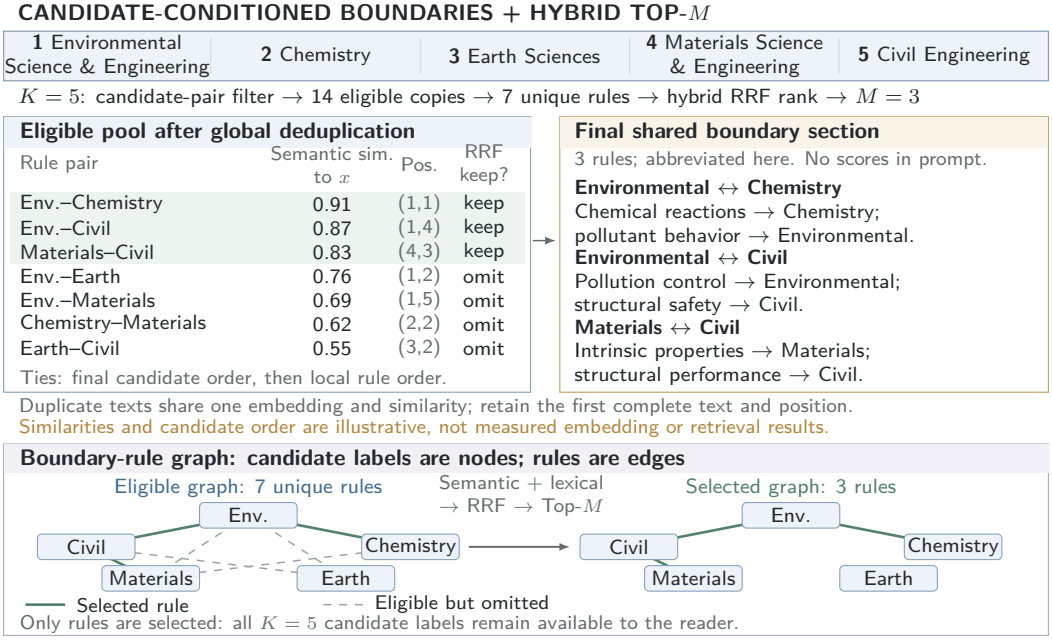

Figure 2: **Candidate-conditioned boundary retrieval** ($K = 5$, $M = 3$). Fourteen eligible copies yield seven unique rules after deduplication. The table shows illustrative semantic similarities and first-occurrence positions; selection combines semantic and lexical ranks using RRF. The graphs retain all five candidate labels while selecting three rules for shared reader context. Scores and candidate order are schematic, not measured results.

same embedding space, and labels are ranked by cosine similarity. Boundary rules are excluded from label embeddings because they describe contrasts between multiple classes and often contain negative or exclusionary semantics, making them unsuitable as positive semantic representations of individual labels.

In parallel, lexical retrieval matches the input against label names and the LLM-generated keywords w_i. These keywords provide direct access to terminology and alternative expressions that may not be captured reliably by semantic similarity alone. They are used only for retrieval and are not included in the final classification prompt, reducing redundant context and token usage.

We combine the semantic and lexical rankings using reciprocal-rank fusion (RRF) (Cormack et al., 2009):

$$q(i \mid x) = \frac{1}{c + \rho_{\text{sem}}(i \mid x)} + \frac{\lambda}{c + \rho_{\text{lex}}(i \mid x)}, \qquad \mathcal{C}_K = \text{TopK}_{y_i \in \mathcal{Y}}\, q(i \mid x), \tag{2}$$

where ρ_{sem} and ρ_{lex} are the ranks from semantic and lexical retrieval, respectively, $c > 0$ is the RRF rank constant, and λ controls the contribution of lexical retrieval. The resulting $\mathcal{C}_K$ contains the Top-K candidate labels passed to the boundary-retrieval stage.

3.4 Candidate-Conditioned Boundary Retrieval

The retrieved Top-K labels may be connected by multiple boundary rules, but many of these distinctions may be irrelevant to a particular input. Including all available rules can introduce distracting information and dilute the boundary evidence most relevant to the current classification decision. SLKR therefore performs a second retrieval stage to select only the boundary rules most relevant to the current input.

Given the candidate set $\mathcal{C}_K$, we first restrict the rule pool to boundaries whose two endpoints are both candidate labels:

$$\mathcal{B}(\mathcal{C}_K) = \{b_{ij} \mid y_i, y_j \in \mathcal{C}_K,\ i \neq j\}. \tag{3}$$

Algorithm 1 SLKR: Offline augmentation and two-stage retrieval

Require: Label inventory $\mathcal{Y}$, task description t, flagship LLM F, encoder h, lightweight LLM g, budgets K, M

Offline: $\mathcal{D} \leftarrow F_{\text{sem}}(\mathcal{Y}, t)$; $\mathcal{W} \leftarrow F_{\text{lex}}(\mathcal{Y}, t)$; $\mathcal{B} \leftarrow F_{\text{bnd}}(\mathcal{Y}, t)$
Construct and freeze $\mathcal{K} \leftarrow (\mathcal{D}, \mathcal{W}, \mathcal{B})$
Online: $\mathcal{C}_K \leftarrow \text{HybridRetrieve}(x, \mathcal{D}, \mathcal{W}, h, K)$ ▷ Top-K labels
$\mathcal{B}(\mathcal{C}_K) \leftarrow \{b_{ij} \in \mathcal{B} \mid y_i, y_j \in \mathcal{C}_K,\ i \neq j\}$
$\mathcal{B}_{K,M}(x) \leftarrow \text{HybridRetrieve}(x, \mathcal{B}(\mathcal{C}_K), h, M)$ ▷ Top-M rules
$\hat{y} \leftarrow g(x, \{(y_i, d_i) \mid y_i \in \mathcal{C}_K\}, \mathcal{B}_{K,M}(x))$
return $\hat{y}$

Rules involving labels outside $\mathcal{C}_K$ are excluded because they do not help distinguish among the retrieved candidates.

We then apply hybrid retrieval over the eligible boundary rules. Semantic retrieval measures the similarity between the input x and each complete boundary rule using the same lightweight encoder h, while lexical retrieval captures direct terminology overlap between the input and the rules. As in label retrieval, the two rankings are combined using RRF, and the Top-M boundary rules are retained:

$$\mathcal{B}_{K,M}(x) = \text{TopM}_{b \in \mathcal{B}(\mathcal{C}_K)}\, q_{\text{bnd}}(b \mid x), \tag{4}$$

where q_{bnd} denotes the fused boundary-retrieval score. This input-conditioned selection allows different rules between the same candidate labels to be emphasized for different inputs, reducing irrelevant distinctions while keeping the final context compact.

Figure 2 illustrates this process. Label retrieval first defines the candidate subspace; only boundaries connecting candidate labels are then considered, and the most input-relevant rules are selected for final classification.

3.5 Lightweight Classification

After the two retrieval stages, the final classification is performed by the lightweight LLM g. The model receives the input text x, the names and definitions of the retrieved Top-K candidate labels, and the selected Top-M boundary rules:

$$\hat{y} = g(x, \{(y_i, d_i) : y_i \in \mathcal{C}_K\}, \mathcal{B}_{K,M}(x))\,. \tag{5}$$

The retrieved definitions provide the semantic meaning of the candidate labels, while the boundary rules provide input-relevant guidance for distinguishing potentially confusable candidates. The lightweight LLM then outputs a single predicted label $\hat{y} \in \mathcal{C}_K$. Retrieval scores and keyword lists are used only during retrieval and are not included in the final prompt, avoiding redundant context and reducing token usage.

3.6 Amortized Cost Analysis

A key motivation of SLKR is that expensive flagship-model computation depends on the label space rather than the number of instances to be classified. Let $N = |\mathcal{Y}|$ denote the number of labels and T the number of input instances. Semantic-definition and keyword generation require $\mathcal{O}(N)$ label-level augmentation, while pairwise boundary construction is upper-bounded by $\mathcal{O}(N^2)$. Thus, the one-time flagship construction cost is $C_{\text{offline}} = \mathcal{O}(N^2)$ in the worst case and, importantly, is independent of T.

After construction, each input requires only retrieval and one lightweight-LLM call. Let c_{online} denote the average online cost per instance. The total and amortized per-instance costs are

$$\begin{aligned} C_{\text{SLKR}}(T) &= C_{\text{offline}} + T c_{\text{online}}, \\ \bar{C}_{\text{SLKR}}(T) &= \frac{C_{\text{offline}} + T c_{\text{online}}}{T} = \frac{C_{\text{offline}}}{T} + c_{\text{online}}. \end{aligned} \tag{6}$$

For a fixed label inventory, the one-time flagship cost is distributed across an increasing number of inputs. Consequently,

$$\lim_{T \to \infty} \frac{C_{\text{offline}}}{T} = 0, \qquad \lim_{T \to \infty} \bar{C}_{\text{SLKR}}(T) = c_{\text{online}}. \tag{7}$$

Table 2: Datasets, label counts, evaluation splits, and roles.

Dataset	Labels	Split	Role
WOS-name-143	143	28,191 train; 9,427 dev; 9,367 test	Scientific documents; domain overlap
BANKING77	77	10,003 train; 3,080 test	Fine-grained banking intents
CLINC150	150	15,000 train; 3,000 dev; 4,500 test	In-scope intent classification
Book-38	38	540 train; 179 test (719 total)	Multilingual books; real-world industrial classification

Thus, at scale, the per-instance contribution of flagship augmentation vanishes, leaving retrieval and lightweight inference as the recurring cost. In contrast, direct flagship classification requires a flagship-model call for every input, so its expensive inference cost grows linearly with T.

4 Experimental Setup

We evaluate SLKR on classification quality, context efficiency, and amortized inference cost. We additionally study the contribution of boundary rules through ablation.

4.1 Datasets

We evaluate SLKR on three public benchmarks and one real-world industrial dataset, summarized in Table 2. **WOS-46985** (Kowsari et al., 2017; 2018) contains scientific abstracts organized in a hierarchical taxonomy; we use the fine-grained Level-2 research areas as classification labels, resulting in 143 classes. **BANKING77** (Casanueva et al., 2020) contains 77 fine-grained banking intents with substantial semantic overlap between classes. **CLINC150** (Larson et al., 2019) contains 150 intent classes spanning multiple domains; we use its in-scope examples for classification.

Book-38 is a real-world industrial dataset for large-scale book classification. It contains 719 multilingual books categorized into 38 broad disciplines, with 540 training and 179 test instances. Compared with the public benchmarks, Book-38 contains longer and more heterogeneous documents across a broader range of subject areas, providing a complementary evaluation in a practical data-curation setting.

4.2 Baselines and Implementation

We compare SLKR against representative zero-shot, few-shot, and retrieval-based classification methods. **Label-only ZS** directly prompts the lightweight LLM with the label inventory. **Gen-Z** (Kumar et al., 2024) augments each label with a semantic description for zero-shot classification. **1-shot/label** provides one labeled demonstration per class, while **Retrieved ICL** retrieves input-relevant labeled demonstrations from the training set. **SLKR w/o Boundary** uses the same label-retrieval pipeline as SLKR but removes boundary rules ($M = 0$), serving as an ablation of boundary retrieval. The full **SLKR** uses $K = 15$ candidate labels and retrieves up to $M = 5$ boundary rules.

All retrieval-based methods use Qwen3-Embedding-0.6B (Zhang et al., 2025) as the embedding model. We evaluate two lightweight readers from different model families, Qwen3.5-4B (Qwen Team, 2026) and Ministral-3-3B (Liu et al., 2026). The offline label knowledge used by SLKR is constructed with GPT-5.6 Sol (OpenAI, 2026) and frozen across online experiments. Importantly, knowledge construction uses only the label inventory and task description, with no access to training, development, or test instances or classifier feedback. In contrast, few-shot and retrieved ICL baselines use labeled examples from the training split. Each experiment is repeated with three random seeds, and we report the mean performance across runs.

Candidate budget. We use $K = 15$ for SLKR across all datasets. As shown in Table 3, Recall@15 exceeds 95% on all four datasets, indicating that a relatively small candidate set retains the correct label for the vast majority of inputs. We therefore fix $K = 15$ for all subsequent experiments.

Table 3: First-stage label retrieval recall (%).

Dataset	R@3	R@5	R@10	R@15
BANKING77	74.74	83.51	94.29	97.18
CLINC150	86.24	92.38	96.47	97.47
WOS-143	79.49	87.81	94.45	95.90
Book-38	73.74	81.38	97.39	98.32

4.3 Boundary Retrieval Ablation

We evaluate boundary retrieval under four settings while fixing the candidate budget at $K = 15$. **No Boundary** ($M = 0$) removes boundary rules entirely. **All Boundaries** includes all eligible rules connecting the retrieved candidate labels. **Random** $M = 5$ randomly selects five eligible boundary rules, providing a rule-count-matched control. **Hybrid Top-**$M = 5$, our default setting, selects the five most input-relevant rules using the hybrid boundary retriever. These settings isolate the effect of boundary knowledge and whether input-conditioned rule selection provides additional benefit over including or randomly selecting boundary rules.

4.4 Metrics

We evaluate classification performance using accuracy and macro-F1. For the first-stage label retriever, we report Recall@K, which measures whether the gold label is retained among the Top-K candidates. Context efficiency is measured by the average number of input tokens supplied to the lightweight reader, while boundary-retrieval experiments additionally report the average number of tokens contributed by selected boundary rules. For cost evaluation, we separately account for the one-time flagship knowledge-construction cost and the recurring online inference cost, following the amortized analysis in Section 3.6.

5 Results

Main results. Table 4 shows that SLKR achieves the best results on all four datasets with Qwen3.5-4B and on three of four datasets with Ministral-3-3B; retrieved ICL is slightly better on CLINC150 with Ministral. The gains are substantially larger for the weaker reader, suggesting that externalized label knowledge can compensate for limited model capacity. SLKR also compares favorably with few-shot methods despite requiring no task-specific labeled examples, while using only 1.4k input tokens on average compared with 4.7k for 1-shot-per-label prompting. Together, these results support the intended division of labor: expensive label understanding is performed offline, while lightweight models can reuse this knowledge with compact online context.

Boundary retrieval. Table 5 shows that explicit boundary knowledge generally improves over definitions alone, but selecting relevant rules is important. Random $M = 5$ consistently underperforms Hybrid Top-$M = 5$, indicating that the gain does not come simply from adding more rule text. Hybrid retrieval uses only 0.35k boundary tokens, compared with 2.51k when all boundaries are included, while retaining similar performance on the public benchmarks. On the more heterogeneous Book-38 dataset, selective retrieval performs substantially better than including all boundaries, suggesting that filtering irrelevant distinctions becomes particularly useful when class boundaries and document content are more diverse.

Flagship and cost comparison. Table 6 compares SLKR with direct GPT-5.6 Sol classification on BANKING77. Direct flagship inference achieves higher classification quality (86.01/85.30 accuracy/macro-F1 versus 77.45/76.63), establishing a clear quality–cost trade-off. However, its inference cost is incurred for every input, whereas SLKR pays the flagship cost only once to construct reusable label knowledge and performs online classification with Qwen3.5-4B. The resulting recurring online cost is substantially lower ($0.0596 versus $4.111 per 1k inputs). As the workload grows, the fixed construction cost is amortized across inputs, consistent with the analysis in Section 3.6.

Table 4: Classification accuracy / macro-F1 (%). Bold marks the best results within each model. Input tokens are averaged across the three public datasets.

Model	Method	WOS-143	BANKING77	CLINC150	Book-38	Avg. Input Tokens
Qwen3.5-4B	Label only – ZS	57.10 / 56.54	68.77 / 66.99	80.36 / 78.94	64.25 / 57.04	0.9k
	Gen-Z (Label + definition)	57.37 / 57.06	72.95 / 71.58	85.64 / 84.66	61.45 / 60.81	2.7k
	1-shot / label	58.67 / 57.88	74.84 / 73.86	89.07 / 88.66	68.16 / 65.57	4.7k
	Retrieved ICL	59.67 / 57.92	75.29 / 74.40	89.24 / 88.92	67.60 / 68.08	1.2k
	SLKR-Desc ($M = 0$)	59.92 / 57.67	77.05 / 75.99	89.13 / 88.57	66.85 / 66.24	1.0k
	SLKR ($M = 5$)	**60.17 / 58.20**	**77.45 / 76.63**	**89.81 / 89.33**	**70.58 / 70.25**	1.4k
Ministral-3-3B	Label only – ZS	29.04 / 29.76	61.17 / 59.96	72.64 / 71.08	39.29 / 35.74	0.9k
	Gen-Z (Label + definition)	34.49 / 35.55	65.91 / 64.79	81.36 / 80.35	46.37 / 44.87	2.7k
	1-shot / label	42.05 / 42.00	71.62 / 70.28	85.31 / 84.69	54.19 / 46.62	4.7k
	Retrieved ICL	46.86 / 45.76	70.26 / 68.59	**86.07 / 85.60**	54.75 / 47.65	1.2k
	SLKR-Desc ($M = 0$)	52.93 / 51.31	67.76 / 66.70	84.00 / 83.22	53.07 / 49.93	1.0k
	SLKR ($M = 5$)	**55.83 / 53.75**	**73.44 / 72.81**	85.69 / 85.07	**58.66 / 52.29**	1.4k

Table 5: Boundary retrieval ablation at $K = 15$. Each cell reports accuracy (top) and macro-F1 (bottom), in %. Boundary tokens denote the average per-input boundary context.

Boundary policy	WOS-143	BANKING77	CLINC150	Book-38	Boundary tokens
No boundary ($M = 0$)	59.92 ± 0.07 57.67 ± 0.07	77.05 ± 0.00 75.99 ± 0.01	89.13 ± 0.03 88.57 ± 0.03	66.85 ± 0.32 66.24 ± 1.62	0
All boundaries	$\mathbf{60.36} \pm 0.08$ $\mathbf{58.34} \pm 0.04$	$\mathbf{77.47} \pm 0.17$ $\mathbf{76.82} \pm 0.20$	$\mathbf{89.95} \pm 0.25$ $\mathbf{89.44} \pm 0.25$	67.04 ± 0.00 64.71 ± 0.18	2.51k
Random $M = 5$	58.72 ± 0.35 56.92 ± 0.31	76.69 ± 0.06 75.78 ± 0.09	89.41 ± 0.18 88.88 ± 0.17	66.48 ± 0.56 62.55 ± 2.04	0.36k
Hybrid Top-$M = 5$	60.17 ± 0.05 58.20 ± 0.04	77.45 ± 0.02 76.63 ± 0.02	89.81 ± 0.03 89.33 ± 0.03	$\mathbf{70.58} \pm 0.32$ $\mathbf{70.25} \pm 0.17$	0.35k

Table 6: Flagship–lightweight cost comparison on BANKING77. Amortized cost is computed over three complete BANKING77 test-set runs (9240 classifications in total). Costs are reported in USD.

Method	Builder	Reader	Accuracy / F1	Build Cost	Online Cost / 1k	Amortized Cost / 1k
Direct Flagship	None	GPT-5.6-sol	86.01 / 85.30	0	4.111	4.111
SLKR	GPT-5.6-sol	Qwen-3.5-4B	77.45 / 76.63	3.4	0.0596	0.4276

6 Conclusion

We presented SLKR, a label-centric retrieval framework for zero-shot text classification that externalizes expensive label understanding into reusable knowledge constructed by a flagship LLM. Rather than retrieving documents or labeled demonstrations, SLKR retrieves candidate-label knowledge and input-relevant boundary rules, enabling lightweight LLMs to classify with compact contexts and without task-specific labeled examples. Across three public benchmarks and a real-world multilingual book-classification dataset, SLKR consistently improves lightweight zero-shot classification and compares favorably with few-shot and retrieved-ICL baselines across two model families. Boundary ablations further show that selecting a small number of relevant rules can retain the benefit of much larger boundary contexts while avoiding irrelevant distinctions. By shifting flagship computation from per-instance inference to reusable offline label knowledge, SLKR provides a practical path toward scalable and cost-efficient classification in large-scale cold-start settings.

AI Use Statement

Generative AI tools were used to polish the writing of this paper and to assist with code development and debugging. All AI-assisted content and code were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Vicente J. Bermejo, Andrés Gago, Ramiro H. Gálvez, and Nicolás Harari. LLMs outperform outsourced human coders on complex textual analysis. *Scientific Reports*, 15(1):40122, 2025. doi: 10.1038/s41598-025-23798-y. URL https://www.nature.com/articles/s41598-025-23798-y.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In Tsung-Hsien Wen, Asli Celikyilmaz, Zhou Yu, Alexandros Papangelis, Mihail Eric, Anuj Kumar, Iñigo Casanueva, and Rushin Shah (eds.), *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 38–45, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlp4convai-1.5. URL https://aclanthology.org/2020.nlp4convai-1.5/.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584836. doi: 10.1145/1571941.1572114. URL https://doi.org/10.1145/1571941.1572114.

Lingyu Gao, Debanjan Ghosh, and Kevin Gimpel. The benefits of label-description training for zero-shot text classification. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13823–13844, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.853. URL https://aclanthology.org/2023.emnlp-main.853/.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. doi: 10.1073/pnas.2305016120. URL https://doi.org/10.1073/pnas.2305016120.

Manish Gupta, Chaitanya Giri, and Jayasimha Talur. From confusion to clarity: Confusion-aware retrieval and knowledge injection for text classification. *arXiv preprint arXiv:2609.01564*, September 2026. doi: 10.48550/arXiv.2609.01564. URL https://arxiv.org/abs/2609.01564v1. Version 1; authors report EMNLP 2026 Industry Track.

Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. HDLTex: Hierarchical deep learning for text classification. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 364–371. IEEE, December 2017. doi: 10.1109/ICMLA.2017.0-134. URL https://doi.org/10.1109/ICMLA.2017.0-134.

Kamran Kowsari, Donald Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew Gerber, and Laura Barnes. Web of Science dataset. Mendeley Data, March 2018. URL https://data.mendeley.com/datasets/9rw3vkcfy4/6. Version 6.

Sachin Kumar, Chan Young Park, and Yulia Tsvetkov. Gen-Z: Generative zero-shot text classification with contextualized label descriptions. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/af7cc9e2366b8f8837a6ef339810277a-Abstract-Conference.html.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1311–1316, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1131. URL https://aclanthology.org/D19-1131/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian

Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Clémence Lanfranchi, Corentin Barreau, Cyprien Courtot, Daniele Grattarola, Darius Dabert, Diego de las Casas, Elliot Chane-Sane, Faruk Ahmed, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet, Georgii Novikov, Guillaume Kunsch, Guillaume Lample, Guillaume Martin, Gunshi Gupta, Jan Ludziejewski, Jason Rute, Joachim Studnia, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Karmesh Yadav, Khyathi Chandu, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Maarten Buyl, Margaret Jennings, Marie Pellat, Mark Prins, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mia Chiquier, Michel Schimpf, Nathan Grinsztajn, Neha Gupta, Nikhil Raghuraman, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Patrick von Platen, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Quentin Torroba, Romain Sauvestre, Roman Soletskyi, Rupert Menneer, Sagar Vaze, Samuel Barry, Sanchit Gandhi, Siddhant Waghjale, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thiziri Nait Saada, Thomas Chabal, Thomas Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Bewley, Tom Edwards, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vincent Maladière, Virgile Richard, Wassim Bouaziz, Wen-Ding Li, William Marshall, Xinghui Li, Xinyu Yang, Yassine El Ouahidi, Yihan Wang, Yunhao Tang, and Zaccharie Ramzi. Ministral 3. *arXiv preprint arXiv:2601.08584*, January 2026. doi: 10.48550/arXiv.2601.08584. URL https://arxiv.org/abs/2601.08584.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In Eneko Agirre, Marianna Apidianaki, and Ivan Vulić (eds.), *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.deelio-1.10. URL https://aclanthology.org/2022.deelio-1.10/.

Zhenyi Lu, Jie Tian, Wei Wei, Xiaoye Qu, Yu Cheng, Wenfeng Xie, and Dangyang Chen. Mitigating boundary ambiguity and inherent bias for text classification in the era of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 7841–7864, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.467. URL https://aclanthology.org/2024.findings-acl.467/.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. Text classification using label names only: A language model self-training approach. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9006–9017, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.724. URL https://aclanthology.org/2020.emnlp-main.724/.

OpenAI. GPT-5.6 Sol model. OpenAI API documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.),

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2655–2671, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.191. URL https://aclanthology.org/2022.naacl-main.191/.

Yubo Wang, Haoyang Li, Fei Teng, and Lei Chen. GORAG: Graph-based online retrieval augmented generation for dynamic few-shot social media text classification. In *Proceedings of the ACM Web Conference 2026*, WWW '26, pp. 9148–9159. Association for Computing Machinery, 2026. ISBN 979-8-4007-2307-0. doi: 10.1145/3774904.3793011. URL https://doi.org/10.1145/3774904.3793011.

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. X-class: Text classification with extremely weak supervision. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3043–3053, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.242. URL https://aclanthology.org/2021.naacl-main.242/.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, June 2025. doi: 10.48550/arXiv.2506.05176. URL https://arxiv.org/abs/2506.05176.

A Additional Method Details

A.1 Independent Label-Knowledge Construction

SLKR constructs three complementary types of label-side knowledge with a flagship LLM: semantic definitions, retrieval keywords, and pairwise boundary rules. As described in Section 3.2, the three augmentation branches are independent. Each branch receives the original label inventory and task description, without using outputs from the other branches or task-specific labeled examples.

Semantic completion expands each label into a concise definition describing its semantic scope and representative topics. Keyword generation produces label-specific terms that are used for lexical candidate retrieval. Boundary construction identifies potentially confusable label pairs and produces natural-language rules describing distinctions between them.

After construction, the three resources are combined into the label-side knowledge base and frozen for online inference. The one-time flagship construction is therefore performed over the label space rather than separately for every input instance.

A.2 Hybrid Candidate-Label Retrieval

The first online stage retrieves candidate labels using both semantic and lexical signals. A frozen lightweight encoder maps the input and augmented label definitions into the same embedding space, and semantic ranking is based on cosine similarity. In parallel, lexical retrieval uses label names and LLM-generated keywords.

The semantic and lexical rankings are combined using reciprocal-rank fusion (RRF). Boundary rules are excluded from the individual label embeddings because they describe contrasts between multiple labels rather than positive semantic representations of a single class. Generated keywords are used for retrieval but are not included in the final lightweight-LLM classification context.

The main experiments use a candidate budget of $K = 15$ for all datasets. Table 3 in the main paper reports Recall@15 of 97.18% on BANKING77, 97.47% on CLINC150, 95.90% on WOS-name-143, and 98.31% on Book-38.

A.3 Candidate-Conditioned Boundary Retrieval

After candidate-label retrieval, SLKR restricts the boundary-rule pool to rules whose two label endpoints both occur in the retrieved Top-K candidate set. Rules involving labels outside the candidate set are excluded.

The eligible boundary rules are then retrieved using hybrid semantic and lexical signals. Semantic retrieval measures similarity between the input and each complete boundary rule using the same lightweight encoder, while lexical retrieval captures direct terminology overlap. The two rankings are combined using RRF, and the Top-M boundary rules are retained.

The main configuration uses $K = 15$ candidate labels and up to $M = 5$ boundary rules. Boundary retrieval changes the decision guidance supplied to the reader; it does not change the candidate-label set itself.

A.4 Lightweight Classification

The lightweight LLM receives the input text, the names and definitions of the retrieved candidate labels, and the selected boundary rules. Retrieved definitions provide semantic information about the candidate labels, while the boundary rules provide input-relevant guidance for distinguishing potentially confusable candidates. Retrieval scores and keyword lists are not included in the final classification context.

The lightweight LLM performs a single online classification call and predicts one label from the retrieved candidate set.

B Additional Dataset Details

B.1 Public Benchmarks

We evaluate on three public benchmarks. WOS-46985 contains scientific abstracts organized in a hierarchical taxonomy; the experiments use the fine-grained Level-2 research-area names as classification labels, resulting in the WOS-name-143 task. BANKING77 contains 77 fine-grained banking intents. CLINC150 contains 150 intent classes spanning multiple domains, and the experiments use its in-scope examples.

The dataset splits used in the experiments are:

Table 7: Dataset splits used in the experiments.

Dataset	Labels	Split
WOS-name-143	143	28,191 train; 9,427 dev; 9,367 test
BANKING77	77	10,003 train; 3,080 test
CLINC150	150	15,000 train; 3,000 dev; 4,500 test
Book-38	38	540 train; 179 test

SLKR knowledge construction uses only the label inventory and task description, with no access to training, development, or test instances or classifier feedback. In contrast, the few-shot and retrieved-ICL baselines use labeled examples from the training split.

B.2 Book-38

Book-38 is a real-world industrial dataset for large-scale book classification. It contains 719 multilingual books categorized into 38 broad disciplines, with 540 training and 179 test instances. Compared with the public benchmarks, Book-38 contains longer and more heterogeneous documents across a broader range of subject areas and provides a complementary evaluation in a practical data-curation setting.

C Additional Experimental Details

C.1 Models

All retrieval-based methods use Qwen3-0.6B as the embedding model. We evaluate two lightweight readers from different model families: Qwen3.5-4B and Ministral-3-3B. The offline label knowledge used by SLKR is constructed with GPT-5.6 Sol and frozen across the online experiments.

Knowledge construction uses only the label inventory and task description and does not access training, development, or test instances or classifier feedback. Few-shot and retrieved-ICL baselines use labeled examples from the training split. Each experiment is repeated with three random seeds, and the main paper reports mean performance across runs.

C.2 Candidate Budget

We use $K = 15$ for SLKR across all four datasets. The first-stage retrieval results reported in Table 3 of the main paper show that Recall@15 exceeds 95% on every dataset. We therefore fix $K = 15$ for the subsequent classification and boundary-retrieval experiments.

C.3 Boundary-Retrieval Ablation

The boundary-retrieval ablation fixes the candidate budget at $K = 15$ and compares four settings:

1. **No Boundary ($M = 0$):** boundary rules are removed.
2. **All Boundaries:** all eligible rules connecting retrieved candidate labels are included.
3. **Random $M = 5$:** five eligible boundary rules are randomly selected.
4. **Hybrid Top-$M = 5$:** the five most input-relevant eligible boundary rules are selected using hybrid boundary retrieval.

These settings separate the effect of explicit boundary knowledge from the effect of input-conditioned boundary selection. As reported in Table 5, Hybrid Top-$M = 5$ uses approximately 0.35k boundary tokens per input, compared with 2.51k for All Boundaries. Random $M = 5$ provides a rule-count-matched control.

C.4 Evaluation Metrics

Classification performance is evaluated using accuracy and macro-F1. First-stage label retrieval is evaluated using Recall@K, measuring whether the gold label is retained in the Top-K candidates. Context efficiency is measured by the average number of input tokens supplied to the lightweight reader. Boundary-retrieval experiments additionally report the average number of tokens contributed by selected boundary rules.

D Cost Accounting

SLKR separates one-time flagship knowledge-construction cost from recurring online inference cost. Let C_{offline} denote the one-time knowledge-construction cost and c_{online} the average recurring online cost per classified input. For T input instances, the total cost is

$$C_{\text{SLKR}}(T) = C_{\text{offline}} + Tc_{\text{online}},$$

and the corresponding amortized per-instance cost is

$$\bar{C}_{\text{SLKR}}(T) = \frac{C_{\text{offline}} + Tc_{\text{online}}}{T} = \frac{C_{\text{offline}}}{T} + c_{\text{online}}.$$

For a fixed label inventory, the contribution of the one-time construction cost per input decreases as the number of classified inputs increases. Consequently,

$$\lim_{T\to\infty} \frac{C_{\text{offline}}}{T} = 0, \qquad \lim_{T\to\infty} \bar{C}_{\text{SLKR}}(T) = c_{\text{online}}.$$

Direct flagship classification, in contrast, requires a flagship-model inference call for every input, so its inference cost grows linearly with T.