Dynamic Label Name Refinement for Few-Shot Dialogue Intent Classification

Anonymous ACL submission

Abstract

Dialogue intent classification aims to identify the underlying purpose or intent of a user's input in a conversation. Current intent classification systems encounter considerable challenges, primarily due to the vast number of possible intents and the significant semantic overlap among similar intent classes. In this paper, we propose a novel approach to few-shot dialogue intent classification through in-context learning, incorporating dynamic label refinement to address these challenges. Our method retrieves relevant examples for a test input from the training set and leverages a large language model to dynamically refine intent labels based on semantic understanding, ensuring that intents are clearly distinguishable from one another. Experimental results demonstrate that our approach effectively resolves confusion between semantically similar intents, resulting in significantly enhanced performance across multiple datasets compared to baselines. We also show that our method generates more interpretable intent labels, and has a better semantic coherence in capturing underlying user intents compared to baselines.

1 Introduction

Dialogue intent classification identifies the underlying intent or purpose of a user's input in a conversation. It is a key component of task-oriented dialogue systems (Degand and Muller, 2020), enabling accurate understanding of user utterances and generation of appropriate responses. However, current intent classification systems face challenges, particularly in managing a large number of intent classes and resolving semantic ambiguity between similar intents (Sung et al., 2023; Cho et al., 2024; Lu et al., 2024). Recent work explores few-shot learning approaches, including retrieval-augmented methods (Milios et al., 2023; Gao et al., 2024; Abdullahi et al., 2024) and prompt-based techniques (Loukas et al., 2023; Parikh et al., 2023; Zhang et al., 2024; Rodriguez et al., 2024), which enable models to learn from limited examples per intent. While retrieval-augmented methods effectively narrow down candidate intents by retrieving examples similar to the input query, these methods also introduce a critical challenge: the retrieved examples often show significant semantic overlap across different intent categories.

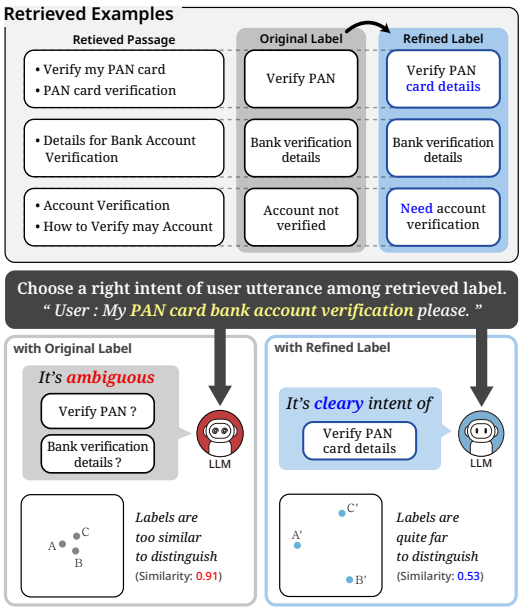


Figure 1: An example illustrating how ambiguous and similar label names can confuse the model, while refined label names enable clearer decision-making.

As shown in Figure 1, even with just three similar intents ('Verify PAN', 'Bank verification details', and 'Account not verified'), the model struggles to make accurate predictions due to their semantic similarity, as indicated by the cosine similarity score of 0.91 at the embedding space. We observe that this high semantic similarity between intent labels makes it challenging for models to distinguish between different intents accurately.

We find that these issues can be mitigated by refining the label names to forms that more distinctly differentiate them from other labels. As illustrated

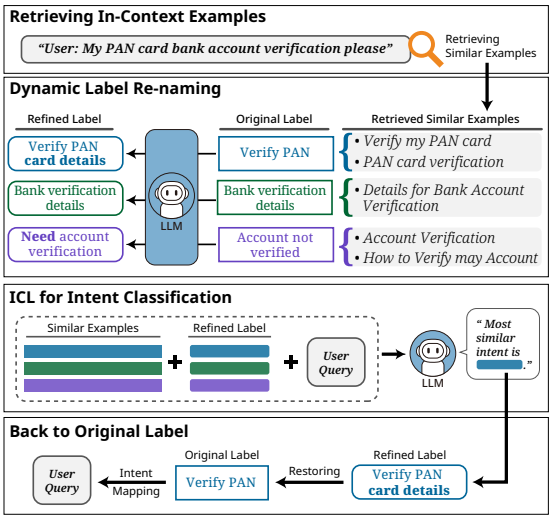


Figure 2: Overall flow of the proposed dynamic label name refinement method for intent classification.

in Figure 1, by mapping the label 'Verify PAN' to a more descriptive form, such as 'Verify PAN card details', it becomes easier to differentiate it from general bank verification intents. This refinement establishes clearer semantic boundaries between intent categories, resulting in more accurate classifications.

In this paper, we present a novel approach that combines dynamic label refinement with similarity-based example selection. Our method involves retrieving semantically similar examples and dynamically refining their intent labels to create more meaningful distinctions between related intents. Through extensive experiments across various model scales and diverse datasets, we achieve significant improvements. Our analysis shows that the improvements are particularly pronounced in datasets with high semantic overlap between intents, with accuracy gains ranging from 2.07% to 7.51% across different model scales.

2 Method

Following the recent works in dialogue intent classification (Milios et al., 2023; Chen et al., 2024; Chandra et al., 2024), our approach leverages retrieval-based in-context learning (ICL) with large language models (LLMs), which has demonstrated effectiveness in tasks involving large label spaces. This approach allows models to dynamically leverage relevant few-shot examples for prediction from the training set.

We introduce a retrieval ICL method for intent classification with dynamic label re-naming, which comprises three steps: (1) retrieving semantically similar examples, (2) refining intent labels using an LLM to generate more descriptive labels, and (3) conducting the final classification with these refined examples.

2.1 Retrieving In-Context Examples

We start by retrieving relevant examples in the dataset for each input query. To retrieve such semantically similar examples, we use a pre-trained SentenceTransformer model (Reimers, 2019). For each input query, we retrieve the top-20 most similar examples. These retrieved examples are then grouped by their original intents to provide comprehensive context for the subsequent steps.

2.2 Dynamic Label Re-naming

Our proposed dynamic label refinement process combines retrieval-based example selection with label generation guided by an LLM, as shown in Figure 2. For each of the specified intent groups, we design tailored instructions for the LLM to analyze these groups and refine the labels accordingly. (see Appendix F for full prompt) Specifically, we design intent label refinement as a process where the model evaluates whether to retain the original label or propose an enhanced version while preserving the domain-specific semantics. During this process, the model assesses the semantic relationship between the label and its associated examples, deciding either to maintain the original intent name or to generate a more descriptive alternative. For instance, as shown in Figure 2, when analyzing examples like "*Verify my PAN card*" and "*Pan card verification*", the model recognizes that the original label '`verify_pan`' could be more descriptive and refines it to '`verify_pan_card_details`' to better capture the specific verification intent.

2.3 ICL for Intent Classification

After obtaining the new labels for each sample, we leverage ICL with the refined labels and examples for final classification. This two-step process where the same LLM both refines the labels and makes the final classification decision helps ensure consistency between the refined semantic understanding and the ultimate intent prediction. Consequently, the overall process involves constructing a prompt that includes: 1) The retrieved examples with their refined intent labels 2) The test query requiring classification 3) Clear instructions for the model to select the most appropriate intent

Dataset	Llama3-8b-inst			Qwen2.5-7b-inst			Qwen2.5-1.5b-inst		
	Raw	**CoT**	**Refined**	**Raw**	**CoT**	**Refined**	**Raw**	**CoT**	**Refined**
HWU64	88.10	87.17 (-0.93)	**89.03** (+0.93)	87.08	86.71 (-0.37)	**88.38** (+1.30)	78.90	78.72 (-0.18)	**80.76** (+1.86)
BANKING77	85.88	85.48 (-0.40)	**87.95** (+2.07)	85.68	87.05 (+1.37)	**87.30** (+1.62)	73.31	72.63 (-0.68)	**77.34** (+4.03)
CLINC150	95.03	95.13 (+0.10)	**95.51** (+0.48)	95.36	95.42 (+0.06)	**95.58** (+0.22)	82.29	81.90 (-0.39)	**84.02** (+1.73)
CUREKART	89.76	89.76 (+0.00)	**91.94** (+2.18)	90.02	89.98 (-0.04)	**90.40** (+0.38)	82.78	82.35 (-0.43)	**84.10** (+1.32)
POWERPLAY11	70.87	67.00 (-3.87)	**76.10** (+5.23)	71.20	70.06 (-1.14)	**74.76** (+3.56)	60.84	58.83 (-2.01)	**63.10** (+2.26)
SOFMATTRESS	82.61	81.10 (-1.51)	**85.40** (+2.79)	83.79	84.46 (+0.67)	**87.40** (+3.61)	73.12	73.70 (+0.58)	**80.63** (+7.51)

Table 1: Performance comparison of our dynamic label refinement approach across different models. **Raw** represents the baseline performance using original intent labels, **CoT** shows results with chain-of-thought prompting, and **Refined** shows the results after applying our dynamic label refinement method.

3 Experiment

3.1 Experimental Setup

Datasets We evaluate our method on two groups of datasets: DialoGLUE benchmark datasets (Mehri et al., 2020) (BANKING77, HWU64, CLINC150) and HINT3 datasets (Arora et al., 2020) (CUREKART, POWERPLAY11, SOFMATTRESS). We provide more details in Appendix A.

Models We conduct experiments with three different sizes of LLMs to evaluate the effectiveness of our approach. We employ *Llama3-8b-inst.* (Dubey et al., 2024), *Qwen2.5-7b-inst.* (Yang et al., 2024), and *Qwen2.5-1.5b-inst.* as our backbone models.

Baselines We compare our approach with the following baselines: 1) **Raw**: A standard in-context learning approach where we retrieve similar examples and perform intent classification using the original intent labels without any refinement. 2) **Chain-of-Thought (CoT)** (Wei et al., 2023): An enhanced prompting method that guides the model to break down the intent classification process into steps, analyzing the user query and retrieved examples before making a prediction.

Implementation Details Following (Milios et al., 2023), we order examples from least to most similar in the prompt, which demonstrated higher accuracy across our datasets. We use this retrieval-based in-context learning setup as our baseline, where the LLM directly performs classification using the original intent labels. While a static label refinement might seem simpler, we opt for dynamic refinement as it enables context-specific label adjustments based on each test query and its retrieved examples. We also confirm this through the experiment in Appendix B.

3.2 Main Results

Table 1 presents a comprehensive analysis of our dynamic label refinement approach across different experimental setups. We structure the investigation around several critical dimensions to clearly demonstrate the effectiveness and implications of our method.

Semantic Disambiguation through Label Refinement We first confirm that while both Raw and CoT approaches show reasonable performance, they often struggle with semantically ambiguous intents. For example, CoT misclassifies "*Please keep delivery service to the pin code 7021*" as `'modify_address'` instead of `'check_pincode'`, and "*Hey I didn't receive the ordered product its incomplete*" as `'refunds_returns_replacements'` instead of `'delay_in_parcel'`. These examples show how structured reasoning often struggles when similar intents have overlapping semantics. Our refinement process tackles this issue by analyzing the semantic relationships between labels and examples, leading to consistent performance improvements over both Raw and CoT baselines across all datasets as shown in Table 1. For example, in BANKING77, this approach achieves notable improvements of +2.07%, +1.62% and +4.03% for Llama3-8b-inst., Qwen2.5-7b-inst., and Qwen2.5-1.5b-inst. respectively, demonstrating the effectiveness of semantic-aware label refinement. The improvements are particularly notable in datasets with complex domain-specific terminology, such as BANKING77, where the model effectively leverages existing semantic information in the labels.

Performance Across Model Scales Our experiments with different model sizes reveal several interesting patterns about the scalability of our approach. The larger models (Llama3-8b-inst. and Qwen2.5-7b-inst.) demonstrate robust baseline

performance with moderate improvements (2.07% and 1.62% on BANKING77, 3.61% and 3.56% on SOFMATTRESS respectively). Notably, even the smaller Qwen2.5-1.5b-inst. Model achieves significant improvements, particularly on domain-specific datasets (+7.51% on SOFMATTRESS, +4.03% on BANKING77), suggesting that our approach effectively enhances performance regardless of model scale. These results demonstrate that our proposed method effectively enhances intent classification performance across various datasets and model architectures.

3.3 Analysis

Semantic Similarity Analysis To validate our hypothesis about semantic relationships between intent labels, we conduct an embedding-based similarity analysis comparing original and refined intent labels across all datasets. Specifically, we leverage the last hidden layer representations from our LLMs to capture the semantic characteristics of each intent label. For each intent, we extracte the final hidden state representation and computed pairwise cosine similarities between these representations within each label set.

Model	Original	Refined
Llama3-8b-inst.	0.86	**0.74**
Qwen2.5-7b-inst.	0.83	**0.80**
Qwen2.5-1.5b-inst.	0.95	**0.91**

Table 2: Average pairwise semantic similarity between original and refined intent labels across various model scales.

We observe that refined labels consistently achieve lower average pairwise similarities than original labels across all datasets and model scales. With Llama3-8b-inst., the average similarity drops from 0.86 for original intent labels to 0.74 for refined labels. Qwen2.5-7b-inst. and Qwen2.5-1.5b-inst. follow a similar trend, with reductions from 0.83 to 0.80 and 0.95 to 0.91, respectively, although the decrease is less significant compared to Llama3-8b-inst. This decrease in semantic overlap directly correlates with improved classification performance. For instance, on the BANKING77 dataset, Llama3-8b-inst. achieves a 2.07% accuracy improvement along with a 0.1 reduction in label similarity. The performance gains become especially noticeable when the model creates more semantically distinct labels, indicating that reducing label overlap enables clearer distinctions between different intents. We also provide the detailed similarity analysis for each dataset in Appendix D.

Dataset	Q2.5-7B		Q2.5-1.5B	
	Q2.5-7B	Q2.5-1.5B	Q2.5-1.5B	Q2.5-7B
HWU64	**88.38**	81.22	80.76	**88.01**
BANKING77	**87.30**	80.81	77.34	**85.95**
CLINC150	**95.58**	87.31	84.02	**95.22**
CUREKART	**90.40**	82.78	84.10	**88.90**
POWERPLAY11	**74.76**	64.10	63.10	**74.43**
SOFMATTRESS	**87.40**	78.30	80.63	**85.08**

Table 3: Performance comparison across different model combinations. The top row means the model used for re-naming, and the second row denotes the models used for classification. Models used: Qwen2.5-7B-inst. (Q2.5-7B) and Qwen2.5-1.5B-inst. (Q2.5-1.5B).

Model Combination To validate the effectiveness of labeling refinement itself of each model, we employ two separate LLMs for intent refinement and classification tasks. We experiment with various combinations of Qwen2.5-7B-inst. and Qwen2.5-1.5B-inst. As in Table 3, using a larger model for refinement followed by a smaller model for classification yields the best performance. For example, using Qwen2.5-7B-inst. for refinement and Qwen2.5-1.5B-inst. for answering achieves notable improvements: +3.31% on CLINC150 and +4.10% on POWERPLAY11 compared to single-model baselines.

Interestingly, even reverse combinations (small model refinement + large model classification) show improvements over the non-refinement baseline, though to a lesser extent. For instance, using Qwen2.5-1.5B-inst. for refinement and Qwen2.5-7B-inst. for answering still achieves improvements on the datasets. This suggests that our label refinement approach is robust across different model scales and configurations.

4 Conclusion

We propose a dynamic label refinement method for few-shot dialogue intent classification that mitigates the issues of significant semantic overlap between intent labels. Using the retrieved examples, we refine labels via LLMs to create more semantically distinct intent categories. Experimental results demonstrate that our method consistently improves performance across multiple datasets for various models. We also confirme that our method reduces semantic similarities between intent labels, creating more distinct and interpretable categories.

Limitations

While our approach demonstrates significant improvements in intent classification performance, it requires additional computational overhead compared to traditional methods. The need to run the model once for label refinement and once for classification - increases the computational cost per query. However, we believe this trade-off is justified by the substantial improvements in classification accuracy, particularly for semantically ambiguous intents.

References

Tassallah Abdullahi, Ritambhara Singh, and Carsten Eickhoff. 2024. Retrieval augmented zero-shot text classification. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '24, page 195–203. ACM.

Gaurav Arora, Chirag Jain, Manas Chaturvedi, and Krupal Modi. 2020. HINT3: Raising the bar for intent detection in the wild. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 100–105, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Manish Chandra, Debasis Ganguly, and Iadh Ounis. 2024. One size doesn't fit all: Predicting the number of examples for in-context learning. *Preprint*, arXiv:2403.06402.

Huiyao Chen, Yu Zhao, Zulong Chen, Mengjia Wang, Liangyue Li, Meishan Zhang, and Min Zhang. 2024. Retrieval-style in-context learning for few-shot hierarchical text classification. *Preprint*, arXiv:2406.17534.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *Preprint*, arXiv:1902.10909.

Jaejin Cho, Rakshith Sharma Srinivasa, Ching-Hua Lee, Yashas Malur Saidutta, Chouchang Yang, Yilin Shen, and Hongxia Jin. 2024. Zero-shot intent classification using a semantic similarity aware contrastive loss and large language model. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10776–10780. IEEE.

Liesbeth Degand and Philippe Muller. 2020. Introduction to the special issue on dialogue and dialogue systems. *Traitement Automatique des Langues*, 61(3):7–15.

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

Lingyu Gao, Aditi Chaudhary, Krishna Srinivasan, Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. 2024. Ambiguity-aware in-context learning with large language models. *Preprint*, arXiv:2309.07900.

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. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Lefteris Loukas, Ilias Stogiannidis, Prodromos Malakasiotis, and Stavros Vassos. 2023. Breaking the bank with ChatGPT: Few-shot text classification for finance. In *Proceedings of the Fifth Workshop on Financial Technology and Natural Language Processing and the Second Multimodal AI For Financial Forecasting*, pages 74–80, Macao. -.

Zhenyi Lu, Jie Tian, Wei Wei, Xiaoye Qu, Yu Cheng, Wenfeng xie, and Dangyang Chen. 2024. Mitigating boundary ambiguity and inherent bias for text classification in the era of large language models. *Preprint*, arXiv:2406.07001.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *Preprint*, arXiv:2009.13570.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. 2023. In-context learning for text classification with many labels. *Preprint*, arXiv:2309.10954.

Soham Parikh, Quaizar Vohra, Prashil Tumbade, and Mitul Tiwari. 2023. Exploring zero and few-shot techniques for intent classification. *Preprint*, arXiv:2305.07157.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Juan A. Rodriguez, Nicholas Botzer, David Vazquez, Christopher Pal, Marco Pedersoli, and Issam Laradji. 2024. Intentgpt: Few-shot intent discovery with large language models. *Preprint*, arXiv:2411.10670.

Mujeen Sung, James Gung, Elman Mansimov, Nikolaos Pappas, Raphael Shu, Salvatore Romeo, Yi Zhang, and Vittorio Castelli. 2023. Pre-training intent-aware encoders for zero- and few-shot intent classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10433–10442, Singapore. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Haode Zhang, Haowen Liang, Liming Zhan, Albert Y. S. Lam, and Xiao-Ming Wu. 2024. Revisit few-shot intent classification with plms: Direct fine-tuning vs. continual pre-training. *Preprint*, arXiv:2306.05278.

A Dataset Details

We evaluate our approach using two different groups of datasets. The first group consists of DialoGLUE benchmark datasets (**BANKING77**: 77 intents focused on banking domain (Casanueva et al., 2020), **HWU64**: 64 intents spanning across 21 domains (Casanueva et al., 2020), **CLINC150**: 150 intents covering 10 domains (Larson et al., 2019)), which are widely used for evaluating task-oriented dialogue systems. For these datasets, we follow the standard 10-shot setting where each intent class has only 10 examples for training, reflecting real-world scenarios where collecting large amounts of labeled data is challenging.

The second group includes HINT3 datasets (Arora et al., 2020) (**CUREKART**: 28 intents in fitness supplements retail domain, **POWERPLAY11**: 59 intents in online gaming domain, **SOFMATTRESS**: 21 intents in mattress products retail domain), which contain real user queries from live chatbots. We exclude the `NO_NODES_DETECTED` label from the test set as it represents out-of-scope queries irrelevant to our task.

B Dynamic vs Static Label Refinement

Dataset	Llama3-8b-inst.			Qwen2.5-1.5b-inst.		
	Baseline	Static	Dynamic	Baseline	Static	Dynamic
HWU64	88.10	84.10 (-4.00)	**89.03** (+0.93)	78.90	74.90 (-4.00)	**80.76** (+1.86)
BANKING77	85.88	81.65 (-4.23)	**87.95** (+2.07)	73.31	69.18 (-4.13)	**77.34** (+4.03)
CLINC150	95.03	87.60 (-7.43)	**95.51** (+0.48)	82.29	83.42 (+1.13)	**84.02** (+1.73)
CUREKART	89.76	85.47 (-4.29)	**91.94** (+2.18)	82.78	**86.27** (+3.49)	84.10 (+1.32)
POWERPLAY11	70.87	68.61 (-2.26)	**76.10** (+5.23)	60.84	55.66 (-5.18)	**63.10** (+2.26)
SOFMATTRESS	82.61	83.00 (+0.39)	**84.60** (+1.99)	73.12	73.91 (+0.79)	**80.63** (+7.51)

Table 4: Performance comparison of baseline, static and dynamic refinement approaches (%). All changes are computed relative to baseline. Best results for each dataset are marked in **bold**, improvements are shown in red, and decreases are shown in blue.

We conduct a detailed comparison between dynamic and static label refinement approaches to validate our method choice. In the static approach, intent labels are refined once before training using all training examples collectively, following the same refinement rules as our dynamic approach. While this might seem advantageous in terms of computational efficiency and consistency, our experimental results demonstrate that the dynamic refinement approach consistently outperforms the static baseline across all datasets and model architectures.

Using Llama3-8b-inst., the dynamic approach achieves +2.07% improvement on BANKING77, while static refinement shows a -4.23% decrease. Similarly on CLINC150, dynamic refinement yields +0.48% improvement, compared to -7.43% with static refinement. Even with smaller models like Qwen2.5-1.5b-inst., dynamic refinement shows consistent gains (+4.03% on BANKING77) while static refinement often leads to performance degradation.

The superior performance of dynamic refinement can be attributed to several factors. Dynamic refinement considers the specific context of each test query and its retrieved examples, allowing for more nuanced label adjustments. By refining labels based on retrieved similar examples, the model can better capture the semantic relationships specific to the current query context. Additionally, the dynamic approach can adjust its refinement strategy based on the semantic similarity patterns observed in the retrieved examples, rather than using fixed refined labels.

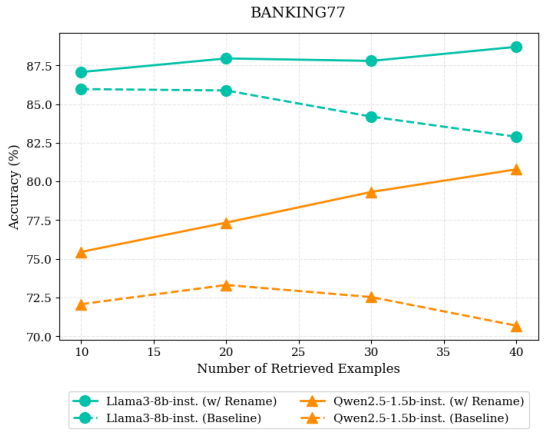
C Impact of the Number of Retrieved Examples

We conduct additional experiments to analyze the impact of the number of retrieved examples on model performance. Figure 3 shows the performance comparison between different model configurations across varying numbers of retrieved examples (10, 20, 30, and 40) on BANKING77 and POWERPLAY11 datasets.

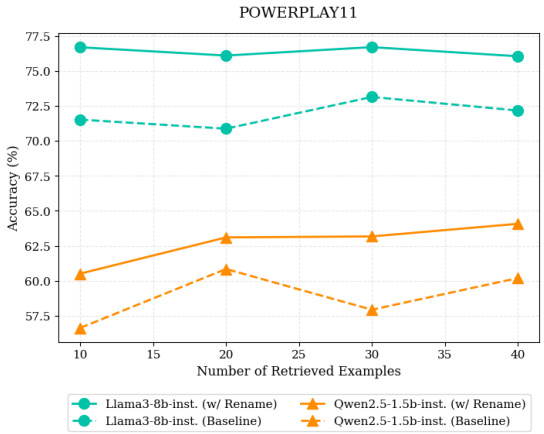
From these results, we observe several key patterns:

- The performance gap between the baseline and refined versions tends to be more pronounced with larger numbers of examples, particularly in BANKING77.
- The larger model (Llama3-8b-inst.) shows more stable performance across different example counts, while the smaller model (Qwen2.5-1.5b-inst.) shows greater variance in performance.
- POWERPLAY11 shows relatively consistent improvement patterns across different example counts, suggesting that the benefits of label refinement are robust across different dataset characteristics.

These findings suggest that our label refinement approach is effective across different numbers of retrieved examples.



BANKING77



POWERPLAY11

Figure 3: Performance comparison with different numbers of retrieved examples. Solid lines represent performance with label refinement, while dashed lines represent baseline performance without refinement.

D Dataset-wise Similarity Analysis

To provide a detailed view of semantic similarity patterns, we analyze the pairwise similarities between intent labels for each dataset. Table 8 shows the average pairwise similarity scores for both original and refined intent labels across different datasets and models.

As shown in Table 6, the reduction in semantic similarity is consistent across all datasets for both models, though the magnitude of reduction varies. For Llama3-8b-inst., POWERPLAY11 shows the largest reduction in similarity (0.156), while CUREKART shows the smallest (0.079). Similarly, for Qwen2.5-1.5b-inst., CLINC150 shows a notable reduction (0.045) while BANKING77 shows a relatively smaller change (0.017). These variations might reflect differences in the initial intent label structures across datasets and the models' ability to refine them effectively.

Dataset	Recall@20	Avg. Intents
HWU64	97.77	7.54
BANKING77	98.93	7.04
CLINC150	99.33	6.31
CUREKART	98.91	4.31
SOFMATTRESS	98.81	5.86
POWERPLAY11	96.44	6.93

Table 5: Retrieval analysis with 20 examples

Dataset	Llama3-8b-inst.		Qwen2.5-7b-inst.		Qwen2.5-1.5b-inst.	
	Original	Refined	Original	Refined	Original	Refined
HWU64	0.835	0.721	0.772	0.760	0.948	0.910
BANKING77	0.889	0.786	0.852	0.838	0.955	0.938
CLINC150	0.834	0.705	0.780	0.764	0.946	0.901
CUREKART	0.881	0.802	0.800	0.833	0.956	0.925
SOFMATTRESS	0.849	0.761	0.752	0.748	0.943	0.911
POWERPLAY11	0.844	0.688	0.838	0.812	0.945	0.903
Mean	0.856	0.744	0.799	0.793	0.949	0.915

Table 6: Detailed semantic similarity analysis across datasets and models. Values represent the average pairwise cosine similarity between all intent labels within each dataset. Lower values indicate more semantically distinct intent categories.

E Semantic vs Generic Label Refinement

While our main experiments focus on semantic-aware refinement that preserves domain context, we also explore an alternative approach using generic identifiers (w/o ori). In this setting, all original intent labels are first replaced with generic identifiers (e.g., Intent_1, Intent_2) before refinement. During refinement, the model generates new labels without any influence from the original intent names.

The effectiveness of each approach varies with model size. Larger models (Llama3-8b-inst. and Qwen2.5-7b-inst.) generally perform better with original name preservation (w ori), particularly in BANKING77 where domain-specific terminology provides valuable semantic context. However, smaller models (Qwen2.5-1.5b-inst.) often show improved performance with generic identifiers (w/o ori), suggesting that they may benefit from the simplified label space. This performance pattern indicates that the choice between preserving or abstracting intent names should consider the model's capacity to leverage domain-specific terminology.

Complete results comparing both approaches

Dataset	Llama3-8b-inst			Qwen2.5-7b-inst			Qwen2.5 -1.5b-inst		
	Raw	Refined w/o ori	Refined w ori	Raw	Refined w/o ori	Refined w ori	Raw	Refined w/o ori	Refined w ori
HWU64	88.10	87.17 (-0.93)	**89.03** (+0.93)	87.08	85.97 (-1.11)	**88.38** (+1.30)	78.90	79.27 (+0.37)	**80.76** (+1.86)
BANKING77	85.88	**87.27** (+1.39)	87.95 (+2.07)	85.68	86.33 (+0.65)	**87.30** (+1.62)	73.31	**79.12** (+5.81)	77.34 (+4.03)
CLINC150	95.03	93.73 (-1.30)	**95.51** (+0.48)	95.36	93.73 (-1.63)	**95.58** (+0.22)	82.29	**86.38** (+4.09)	84.02 (+1.73)
CUREKART	89.76	**91.94** (+2.18)	**91.94** (+2.18)	90.02	**90.40** (+0.38)	**90.40** (+0.38)	82.78	**86.27** (+3.49)	84.10 (+1.32)
POWERPLAY11	70.87	76.05 (+5.18)	**76.10** (+5.23)	71.20	74.43 (+3.23)	**74.76** (+3.56)	60.84	**66.99** (+6.15)	63.10 (+2.26)
SOFMATTRESS	82.61	83.40 (+0.79)	**85.40** (+2.79)	83.79	82.60 (-1.19)	**87.40** (+3.61)	73.12	79.44 (+6.32)	**80.63** (+7.51)

Table 7: Complete performance comparison including refinement without original name preservation (Refined w/o ori). Results show that while both refinement approaches generally improve over the baseline (Raw), preserving original names during refinement (Refined w ori) tends to yield better or comparable results.

across all datasets and models are shown in Table 7.

F Full Prompt Template

F.1 Re-naming Prompt Template

Examples by intent:
Intent 1: account_not_verified
- How to Verify my Account?
- Account Verification
- Need to Verify my account

Intent 2: delete_pan_card
- Pan card remove
- Delete PAN card
- I want to delete my pan card

Intent 3: bank_verification_details
...
Intent 4: pan_verification_failed
...
Rules for intent mapping:
1. If the current intent name accurately represents its examples, map it to itself
2. If the intent name needs improvement, create a new descriptive name that better represents the examples
3. For new names:
- Use lowercase letters only
- Use underscores between words

INTENT MAPPINGS:
account_not_verified ->
delete_pan_card ->
...
pan_verification_failed ->

Table 8: Re-naming prompt template.

The re-naming prompt takes groups of examples organized by their original intent labels and analyzes their semantic meaning. Based on this analysis, it either keeps the original label if it accurately represents the examples, or generates a new more descriptive label while maintaining consistent formatting rules. The prompt enforces lowercase letters and underscores in label names to ensure standardization.

You are an AI assistant specialized in intent classification. Your task is to determine
the single most likely intent of a given query based on the examples provided.
Provide only the name of the most probable intent, without any additional text or explanation.

...

Text: "Details for Bank Account Verification"
Intent: bank_verification_details

Text: "Getting error while verifying PAN Card"
Intent: pan_verification_failure

Text: "My PAN card needs to be verified"
Intent: verify_pan_card_details

Query: "My PAN card bank account verification please"
The top 1 most likely intent is:

Table 9: Classification prompt template.

F.2 Classification Prompt Template

The classification prompt presents example pairs of text queries and their corresponding refined intents to establish the task context. It instructs the model to determine the most likely intent for a new query based solely on these examples. The prompt explicitly requires only the intent name as output, without any additional explanation or text, to ensure consistent and clean predictions.

You are an AI assistant specialized in intent classification. Your task is to determine
the single most likely intent of a given query based on the examples provided.
For each query:
1. Analyze the key elements and meaning
2. Provide an explanation of your reasoning
3. Extract the most likely intent

Text: "Details for Bank Account Verification"
Intent: bank_verification_details

Text: "Getting error while verifying PAN Card"
Intent: pan_verification_failure

Text: "My PAN card needs to be verified"
Intent: verify_pan_card_details

Query: "My PAN card bank account verification please"
Provide your explanation and intent:

Table 10: Chain-of-Thought prompt template.

F.3 Chain-of-Thought Prompt Template

The Chain-of-Thought prompt extends the basic classification prompt by requiring the model to explain its reasoning process. For each query, the model must analyze the key elements, provide reasoning, and then determine the intent. This structured approach aims to help the model make more informed decisions by breaking down the classification process into steps.

G Related Work

G.1 Dialogue Intent Classification

Dialogue intent classification aims to identify users' intentions from natural language utterances. Traditional approaches relied on supervised learning with large labeled datasets (Chen et al., 2019; Larson et al., 2019). The advent of large language models (LLMs) transforms this landscape, enabling effective few-shot learning approaches where only limited labeled data is available (Brown et al., 2020). This shift has been particularly significant as LLMs demonstrate strong few-shot capabilities through in-context learning, reducing the need for extensive labeled datasets (Loukas et al., 2023; Parikh et al., 2023; Chen et al., 2024).

G.2 Retrieval-based In-Context Learning

Retrieval-based approaches have emerged as a powerful paradigm for improving few-shot learning performance. Key developments in this area include: (Milios et al., 2023) propose effective retrieval strategies for in-context learning with many labels, demonstrating significant performance improvements through careful example selection. However, while this approach successfully retrieves semantically similar examples for classification, it introduces new challenges when dealing with intent labels that have high semantic overlap. (Lu et al., 2024) This ambiguity between similar intents creates unnecessary complexity in the classification task, particularly when multiple intents share similar contextual meanings but require different downstream processing. Our work addresses this challenge by introducing a dynamic label refinement approach that helps distinguish between semantically similar intents while maintaining the benefits of retrieval-based example selection.

H Case Study

H.1 Label Refinement Pattern Analysis

Our analysis revealed interesting patterns in how the model refines intent labels, particularly highlighting some suboptimal refinement behaviors:

H.1.1 Verbatim Query-to-Intent Conversion

We observed cases where the model simply converted user queries directly into intent labels:

- Original_text: "I want to return my mattress"
- Refined_intent: `i_want_to_return_my_mattress`

This pattern indicates a potential limitation in our refinement approach where the model sometimes fails to abstract the core intent, instead creating overly specific labels that mirror the input text.

H.1.2 Overly Descriptive Intent Labels

Another pattern emerged where the model generated unnecessarily verbose intent labels:

- Original_intent: `size_customization`
- Refined_intent: `how_can_i_order_a_custom_sized_mattress`

These findings highlight the need for more sophisticated label refinement strategies that maintain a balance between descriptiveness and practical utility.

I Performance comparison across different model combinations.

Dataset	L3-8B		Q2.5-1.5B	
	L3-8B	Q2.5-1.5B	Q2.5-1.5B	L3-8B
HWU64	**89.03**	80.95	80.76	**88.48**
BANKING77	**87.95**	80.09	77.34	**88.47**
CLINC150	**95.51**	86.82	84.02	**94.79**
CUREKART	**91.94**	84.96	84.10	**91.07**
POWERPLAY11	**76.10**	65.69	63.10	**74.11**
SOFMATTRESS	**85.40**	79.44	80.63	**85.38**

Table 11: Performance comparison across different model combinations. The top row shows the model used for re-naming, while the models listed in the second row are used for answering (classification).

Based on Table 3, we conduct additional experiments employing two separate large language models (LLMs) for intent segmentation and classification tasks in order to verify the effectiveness of label segmentation for each model. In particular, for the two models used in the experiment, we report the accuracy of both the model that generates the intents and the model that generates the responses. Consistent with the results shown in Table 3, we find that using a larger model for response generation is effective. Furthermore, we observe that label re-naming, even when performed using labels derived from a smaller model, still yields strong performance.