REIC: RAG-Enhanced Intent Classification at Scale

Ziji Zhang
czhangzi@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Michael Yang
abyang@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Zhiyu Chen
zhiyuche@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Yingying Zhuang
yyzhuang@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Shu-Ting Pi
shutingp@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Qun Liu
qunliu@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Rajashekar Maragoud
maragoud@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Vy Nguyen
nguynvy@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Anurag Beniwal
beanurag@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Abstract

Accurate intent classification is critical for efficient routing in customer service, ensuring customers are connected with the most suitable agents while reducing handling times and operational costs. However, as companies expand their product lines, intent classification faces scalability challenges due to the increasing number of intents and variations in taxonomy across different verticals. In this paper, we introduce **REIC**, a **R**etrieval-augmented generation **E**nhanced **I**ntent **C**lassification approach, which addresses these challenges effectively. REIC leverages retrieval-augmented generation (RAG) to dynamically incorporate relevant knowledge, enabling precise classification without the need for frequent retraining. Through extensive experiments on real-world datasets, we demonstrate that REIC outperforms traditional fine-tuning, zero-shot, and few-shot methods in large-scale customer service settings. Our results highlight its effectiveness in both in-domain and out-of-domain scenarios, demonstrating its potential for real-world deployment in adaptive and large-scale intent classification systems.

CCS Concepts

• **Computing methodologies → Classification and regression trees**; **Natural language processing**.

Keywords

Intent Classification, RAG, ICL, text classification

ACM Reference Format:
Ziji Zhang, Michael Yang, Zhiyu Chen, Yingying Zhuang, Shu-Ting Pi, Qun Liu, Rajashekar Maragoud, Vy Nguyen, and Anurag Beniwal. 2025. REIC: RAG-Enhanced Intent Classification at Scale. In *Proceedings of LLM4ECommerce Workshop at the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (LLM4ECommerce Workshop at KDD '25).* ACM, New York, NY, USA, 7 pages.

1 Introduction

Customer service [3, 4, 14, 16, 17, 28, 30] is critical for modern e-commerce but also one of the most resource-intensive departments. Different agents, either human or model, are trained to handle specific types of customer issues, making precise intent classification, particularly at the issue level, crucial for efficient routing. High issue-oriented intent accuracy ensures that customers are connected with the most suitable agents, reducing unnecessary transfers and lowering handling times. This optimization not only enhances customer satisfaction but also cuts operational costs by streamlining interactions and improving overall service efficiency. For model-based automatic resolvers in chatbot agentic systems [2, 6], the ability to precisely identify user intent is essential for delivering contextually appropriate and solution-oriented responses.

As companies expand their product lines, intent classification faces two key challenges. First, the number of customer intents grows over time, requiring models to adapt to new intents quickly. Second, intent taxonomies can vary across product lines, making it difficult to maintain a unified classification system. For example we organize products into different verticals in e-commerce: with third-party products, customer usually inquire about physical retail orders or consumer accounts, and intents are categorized into three levels from coarse to fine-grained. In contrast, first-party products require more customized customer services due to our proprietary device and digital product offerings. For instance, a customer seeking device troubleshooting may interact with an agent who can access real-time diagnostic information and perform specific troubleshooting steps on the user's behalf. This necessitates a broader set of intent categories to accommodate diverse customer needs, as illustrated in Figure 1. This heterogeneity complicates intent classification, demanding scalable and flexible approaches to ensure accurate routing and efficient customer service. In this work we demonstrate our method using two verticals but it can be easily adapted to more.

In this paper, we propose a novel **R**etrieval-augmented generation **E**nhanced **I**ntent **C**lassification (**REIC**) approach that reduces computational complexity and improves scalability for intent classification. We demonstrate the effectiveness of this approach through extensive experiments on real-world datasets , showing that our method outperforms traditional fine tuning and zero-shot or few-shot methods in large-scale customer service settings. Our results

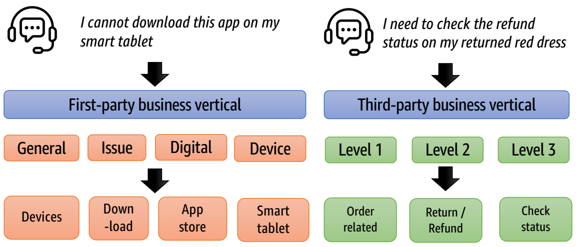

Figure 1: We present the heterogeneous intent structure with examples, highlighting the label hierarchy per vertical.

on both in-domain and out-of-domain intents demonstrate its potential to improve classification accuracy and enable dynamic updates without retraining, making it ideal for industry-scale applications.

2 Related Work

2.1 Intent classification

Early work on intent classification for dialogues often relied on bag-of-words or recurrent models. For example, Schuurmans and Frasincar [21] evaluated various classifiers on a multi-domain intent dataset and found that a simple SVM with hierarchical label taxonomy outperformed deeper LSTM models. With the advance of transformer architectures, researchers began to leverage self-attention and multi-task learning for intent understanding. Ahmadvand et al. [1] introduced a joint intent mapping model that simultaneously classifies high-level intent and maps queries to fine-grained product categories. Wang et al. [27] employed a slowly updated text encoder and global/local memory networks to mitigate catastrophic forgetting and parameter explosion for large-scale intent detection task. Recent work has pushed toward using large pre-trained models and retrieval-based prompting to enable cross-domain and zero/few-shot intent classification. Liu et al. [11] proposed a framework which integrates a fine-tuned XLM-based intent classifier with an LLM to essentially treat multi-turn intent understanding as a zero-shot task. Yu et al. [29] also explored retrieval-based methods for intent classification and slot filling tasks in few-shot settings. Our work adopts similar in-context learning (ICL) setup while focusing on handling large-scale multi-domain intent classification task from industry level applications.

2.2 In-context Learning

The performance of LLM has been significantly enhanced in few-shot and zero-shot NLP tasks through ICL. Recent ICL research focus on how to effectively identify and interpret retrieved context. Guu et al. [7] first showed how to pre-train masked language models with a knowledge retriever in an unsupervised manner. Karpukhin et al. [10] proposed a training pipeline in which retrieval is implemented using dense representations alone and embeddings are learned from a small number of questions and passages with a dual-encoder. Ram et al. [18] considered simple alternatives to only prepend retrieved grounding documents to the input, instead of modifying the LLM architecture to incorporate external information. Similar approaches have proven particularly effective in the application of RAG on dialogue systems [23, 24], specifically goal-oriented and domain-specific dialogs from customer service scenarios [22, 31, 32]. In our work, we utilize ICL in both LLM fine-tuning stage for data generation and at inference-time with an intent candidate retriever.

3 Preliminary

Intent classification for queries is typically framed as a multiclass text classification problem. Specifically, given a customer query $q \in Q$, the goal is to map it to one of the k pre-defined intents $t \in T = \{t_1, ..., t_k\}$ using a model $\mathcal{M}$ so that the predicted intent $\hat{t} = \mathcal{M}(q)$ maximizes the probability of correctly classifying q. Formally, this can be expressed as:

$$\hat{t} = \arg\max_{t \in T} P(t \mid q; \theta) \tag{1}$$

where $P(t \mid q; \theta)$ denotes the probability of intent t given query q, parameterized by θ of the model $\mathcal{M}$.

Flat classification assumes independence among intent labels, which is often unrealistic in practice. Two key challenges arise at industry scale: 1) **Scalability**—a large number of labels makes flat classification inefficient and hard to maintain; 2) **Label correlation**—related intents (e.g., "Order Issue" vs. "Track Order") are treated independently, ignoring useful hierarchical structure. To address this, we adopt hierarchical intent classification, where queries are classified progressively from general to specific intents. This improves both efficiency and accuracy.

Note that for a more generalized setting, intents from different verticals and domains may have entirely different hierarchy and ontology. In Figure 1, we demonstrated some examples of intent hierarchy in our application, which involves customer service query intent detection with two business verticals: **Third-Party** or **3P** business (customer contacting about third-party physical retail orders or consumer accounts) and **First-Party** or **1P** business (customer contacting about first-party digital or device issues). If we use the traditional single-head flattened intent labels, the total intent ontology set size would be at 10^3 level which create major challenges for accurate intent classification. By creating hierarchical intent ontology across different business verticals, each classification head only needs to handle less than 50 intents that are more manageable for language models. In the following sections of this paper, we utilize this intent ontology setup for experiments and comparisons.

4 Method

Adapting Large Language Models (LLMs) to domain-specific intent detection is challenging due to specialized terminology, organizational language, and unique customer scenarios. To address these limitations and enhance the accuracy of customer intent identification, we introduce a novel approach REIC for RAG-Enhanced Intent Classification. REIC consists of three components: index construction, candidate retrieval, and intent probability calculation (as illustrated in Figure 2).

Index Construction. We first construct a dense vector index containing (query, intent) pairs from a held-out annotated dataset. Each query is encoded using a pre-trained sentence transformer model to generate dense vector representations. The corresponding intent labels are stored alongside these embeddings. The intent labels

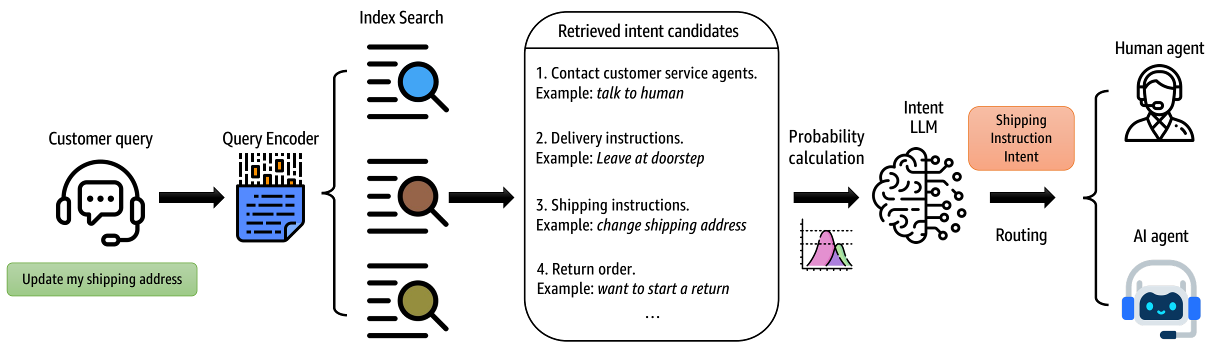

Figure 2: The proposed REIC method maps customer queries to routing intents using vector retrieval and probability estimation.

follow a hierarchical structure with d dimensions which represents different intent domain knowledge and might range from different domains.

Candidate Retrieval. Given a new query q, we first encode it using the same encoder for index construction to obtain its dense vector representation $\mathbf{v}_q$. We then perform approximate nearest neighbor search to retrieve the top-k most similar (query, intent) pairs, denoted as set E. The similarity is computed using cosine distance between the query vector and indexed vectors:

$$sim(q, q_i) = \frac{\mathbf{v}_q \cdot \mathbf{v}_i}{\|\mathbf{v}_q\| \|\mathbf{v}_i\|} \tag{2}$$

where $\mathbf{v}_i$ represents the vector encoding of the i-th indexed query.

Intent Probability. For the retrieved set E, we leverage a fine-tuned LLM $\mathcal{M}$ to perform constrained decoding and calculate the probabilities of the possible intents. Given a prompt template $\mathcal{P}$, the LLM takes as input the instantiated prompt, which includes the original query q and the retrieved (query, intent) pairs as context. For each unique intent t_j in E, we compute:

$$P(t_j | q, E) = \mathcal{M}(\mathcal{P}, q, E)_{t_j} \tag{3}$$

where $\mathcal{M}(\mathcal{P}, q, E)_{t_j}$ represents the model's predicted probability for t_j given the query and retrieved examples.
The final intent classification is determined by selecting the intent with the highest probability:

$$\hat{t} = \arg\max_{t_j \in E} P(t_j | q, E) \tag{4}$$

This approach enables dynamic updates to the intent space by simply adding new (query, intent) pairs to the index, leveraging the in-context learning capabilities of the LLM without requiring model retraining.

The probability-based reranking helps mitigate potential LLM hallucination by grounding predictions in retrieved examples. With tradition greedy decoding, sometimes the LLM might generate intents outside of the given candidate list and cause downstream routing failure. We perform constrained decoding to calculate the probability of each retrieved intent t_j in E, which ensures the success of downstream routing. Given prompt $\mathcal{P}$ with instructions, retrieved candidates E, and customer query q, we append t_j at the end to calculate the total logits from model forward pass $\mathcal{L}_{t_j} = \mathcal{M}(\mathcal{P}(E, q) + t_j)$. Then we mask out the positions of $\mathcal{P}(E, q)$ and accumulate the log probabilities for the intent sequence t_j with length s_j:

$$\mathcal{M}(\mathcal{P}, q, E)_{t_j} = \exp\big(\sum_{t_j} \mathrm{LogSoftmax}(\mathcal{L}_{t_j})/s_j\big) \tag{5}$$

During training, we train the intent LLM $\mathcal{M}$ by minimizing the cross-entropy loss between the predicted and ground-truth intents. During inference, instead of traditional auto-regressive next token decoding, we perform one model forward-pass calculation with a batch size k for top-k intent candidates and get the k probabilities for re-ranking and final intent prediction.

5 Experimental Setup

5.1 Datasets

We applied data anonymization described in Appendix A and the final dataset contains 52,499 training samples with 35,041 **1P Business** queries and 17,458 **3P Business** queries. The test set consists of 3,647 **1P Business** queries and 1,717 **3P Business** queries respectively using random sampling. All of the data samples have incorporated retrieved intent candidates from the retriever. We also performed dataset cleaning in the training set to make sure the true intent is contained in the retrieved list. During inference, we use the actual noisy retrieved list which also relies on the capability of the embedding model.

5.2 Compared Methods

We consider the following baselines:

- **RoBERTa**: We fine-tune RoBERTa-base [12] with multiple classification heads. This adaptation allowed the model to simultaneously categorize utterances across multiple dimensions.
- **Mistral Classification**[1]: We fine-tune a Mistral-7B-v0.3[2] with a sequence classification head. Instead of directly generating output sequences, the model projects the pooled embedding into a space with the same dimension as the number of classes.
- **Claude Zero-shot**: We employ the Claude 3.5 Sonnet model in a zero-shot configuration. To facilitate accurate intent prediction, we craft a comprehensive prompt that explicitly defines each potential intent.

[1]Due to legal concerns, we are not permitted to use non-commercial LLMs like Llama.
[2]https://huggingface.co/mistralai/Mistral-7B-v0.3

Table 1: Intent detection confusion matrix on different business with different methods

Models	3P Business vertical			1P Business vertical			Overall		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
RoBERTa	0.527	0.447	0.483	0.583	0.488	0.531	0.565	0.474	0.516
Mistral Classification	0.215	0.228	0.221	0.301	0.250	0.273	0.269	0.243	0.255
Claude Zero-shot	0.338	0.250	0.287	0.238	0.170	0.199	0.271	0.196	0.227
Claude Few-shot	0.386	0.289	0.331	0.350	0.308	0.328	0.361	0.302	0.329
Claude + RAG	0.473	0.438	0.455	0.415	0.389	0.402	0.434	0.404	0.419
REIC	**0.538**	**0.546**	**0.542**	**0.600**	**0.574**	**0.587**	**0.579**	**0.565**	**0.572**

- **Claude Few-shot**: Similar to **Claude Zero-shot**, we incorporate 20 demonstration examples, with 10 from each vertical, to enhance coverage of diverse intents across different domains.
- **Claude+RAG**: Instead of using a fine-tuned LLM, we employ Claude 3.5 Sonnet as the backbone and incorporate the same set of retrieved candidates as described in Section 4 into the prompt. This comparison allows us to assess whether a smaller fine-tuned LLM can perform competitively against a large foundation model for this task.

5.3 Implementation Details

The LLM component of our REIC approach utilizes a fine-tuned model from Mistral-7B-Instruct-v0.2[3]. The retrieval index is implemented using FAISS [5] for efficient similarity search in high-dimensional space. We experimented with four different retrievers: BM25 [19], MPNet [25], Contriever-MS MACRO [9] and ColBERT-v2 [20]. More implementation details are described in Appendix D.

6 Results

6.1 Intent Detection Ability

To assess REIC's effectiveness in intent detection, we compared it against baseline methods (see §5.2), with results shown in Table 1. The experiments, covering the *3P Business* and *1P Business* verticals, evaluated Precision, Recall, and F1-score.

The results indicate that REIC outperforms standard fine-tuning and prompting-based methods. While fine-tuned models like RoBERTa perform reasonably well, they require extensive retraining when new intents emerge. We hypothesize that the limited performance of the Mistral Classification model stems from its nature as a decoder-only architecture, which may be less effective in extracting the semantic meaning of input query. Prompting-based approaches (Claude Zero-shot and Few-shot) generally underperformed, with Claude Few-shot achieving a maximum F1-score of 0.329 overall. The Claude + RAG method improved performance compared to standalone prompting but remained inferior to our approach by 26.7%.

These observations confirm that the integration of RAG and fine-tuned LLM enables greater flexibility, improved precision, and higher recall rates, making it well-suited for handling diverse and evolving intent spaces in different applications. We further evaluate the impact of different retrievers and the size of retrieval candidates in Appendix B and Appendix C.

6.2 Robustness on Unseen Intents

To evaluate REIC's robustness on unseen intents, we trained our models exclusively on the *3P Business* vertical and tested them on the *1P Business* vertical, simulating a real-world out-of-domain scenario. As illustrated in Figure 1, *1P Business* vertical has 4 intent category with more than 800 unique intent combinations, while the training data used from *3P Business* vertical has 3 intent category with only around 70 unique intents. This out-of-domain scenario helps assess how well REIC generalizes to new, previously unseen intents. The results are summarized in Table 2.

As seen in the table, Claude Zero-shot performs poorly with 0.17 accuracy, while Claude Few-shot improves to 0.308. RAG-based methods, particularly Claude + RAG, outperform Claude Few-shot with 0.389 accuracy, demonstrating the advantage of dynamic retrieval. REIC, though slightly lower than Claude + RAG, still outperforms Claude Few-shot, underscoring its strong performance on both in-domain and out-of-domain intents. Overall, REIC proves robust and adaptable to unseen domains.

Table 2: Out-of-domain intent detection accuracy

Models	3P Business	1P Business	Overall
Claude Zero-shot	0.250	0.170	0.196
Claude Few-shot	0.289	0.308	0.302
Claude + RAG	0.438	**0.389**	**0.404**
REIC	**0.538**	0.283	0.364

7 Conclusion

This paper presents a novel RAG-Enhanced Intent Classification (REIC) method that addresses scalability challenges and the heterogeneity of intent taxonomies in large-scale customer service systems. By incorporating a hierarchical intent classification strategy, REIC significantly reduces computational complexity. Leveraging the RAG technique, our method dynamically integrates contextually relevant retrieved examples, outperforming traditional fine-tuning, as well as zero-shot and few-shot approaches, in intent detection tasks. Additionally, our results demonstrate strong performance on both in-domain and out-of-domain test sets, highlighting its applicability for industry-scale applications.

[3]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

References

[1] Ali Ahmadvand, Surya Kallumadi, Faizan Javed, and Eugene Agichtein. 2020. Jointmap: joint query intent understanding for modeling intent hierarchies in e-commerce search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1509–1512.

[2] Zhiyu Chen, Jason Choi, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2023. Generate-then-Retrieve: Intent-Aware FAQ Retrieval in Product Search. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, Sunayana Sitaram, Beata Beigman Klebanov, and Jason D Williams (Eds.). Association for Computational Linguistics, Toronto, Canada, 763–771. doi:10.18653/v1/2023.acl-industry.73

[3] Zhiyu Chen, Jason Ingyu Choi, Besnik Fetahu, and Shervin Malmasi. 2024. Identifying High Consideration E-Commerce Search Queries. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (Eds.). Association for Computational Linguistics, Miami, Florida, US, 563–572. doi:10.18653/v1/2024.emnlp-industry.42

[4] Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. 2017. Superagent: A customer service chatbot for e-commerce websites. In *Proceedings of ACL 2017, system demonstrations*. 97–102.

[5] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. (2024). arXiv:2401.08281 [cs.LG]

[6] Aman Gupta, Anirudh Ravichandran, Ziji Zhang, Swair Shah, Anurag Beniwal, and Narayanan Sadagopan. 2024. DARD: A multi-agent approach for task-oriented dialog systems. *arXiv preprint arXiv:2411.00427* (2024).

[7] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. arXiv:2002.08909 [cs.CL] https://arxiv.org/abs/2002.08909

[8] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[9] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research* (2022). https://openreview.net/forum?id=jKN1pXi7b0

[10] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. arXiv:2004.04906 [cs.CL] https://arxiv.org/abs/2004.04906

[11] Junhua Liu, Yong Keat Tan, Bin Fu, and Kwan Hui Lim. 2024. LARA: Linguistic-Adaptive Retrieval-Augmentation for Multi-Turn Intent Classification. *arXiv preprint arXiv:2403.16504* (2024).

[12] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[13] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. https://github.com/huggingface/peft.

[14] Hoang Nguyen, Chenwei Zhang, Congying Xia, and Philip Yu. 2020. Dynamic Semantic Matching and Aggregation Network for Few-shot Intent Detection. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 1209–1218. doi:10.18653/v1/2020.findings-emnlp.108

[15] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset. (2016).

[16] Soham Parikh, Mitul Tiwari, Prashil Tumbade, and Quaizar Vohra. 2023. Exploring Zero and Few-shot Techniques for Intent Classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, Sunayana Sitaram, Beata Beigman Klebanov, and Jason D Williams (Eds.). Association for Computational Linguistics, Toronto, Canada, 744–751. doi:10.18653/v1/2023.acl-industry.71

[17] Haode Qi, Lin Pan, Atin Sood, Abhishek Shah, Ladislav Kunc, Mo Yu, and Saloni Potdar. 2021. Benchmarking Commercial Intent Detection Services with Practice-Driven Evaluations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, Young-bum Kim, Yunyao Li, and Owen Rambow (Eds.). Association for Computational Linguistics, Online, 304–310. doi:10.18653/v1/2021.naacl-industry.38

[18] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-Context Retrieval-Augmented Language Models. arXiv:2302.00083 [cs.CL] https://arxiv.org/abs/2302.00083

[19] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[20] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 3715–3734. doi:10.18653/v1/2022.naacl-main.272

[21] Jetze Schuurmans and Flavius Frasincar. 2019. Intent classification for dialogue utterances. *IEEE Intelligent Systems* 35, 1 (2019), 82–88.

[22] Lütfi Kerem Senel, Besnik Fetahu, Davis Yoshida, Zhiyu Chen, Giuseppe Castellucci, Nikhita Vedula, Jason Ingyu Choi, and Shervin Malmasi. 2024. Generative Explore-Exploit: Training-free Optimization of Generative Recommender Systems using LLM Optimizers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 5396–5420. doi:10.18653/v1/2024.acl-long.295

[23] Kurt Shuster, Mojtaba Komeili, Leonard Adolphs, Stephen Roller, Arthur Szlam, and Jason Weston. 2022. Language Models that Seek for Knowledge: Modular Search & Generation for Dialogue and Prompt Completion. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 373–393. doi:10.18653/v1/2022.findings-emnlp.27

[24] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. 2022. BlenderBot 3: a deployed conversational agent that continually learns to responsibly engage. arXiv:2208.03188 [cs.CL] https://arxiv.org/abs/2208.03188

[25] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems* 33 (2020), 16857–16867.

[26] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=wCu6T5xFjeJ

[27] Chengyu Wang, Haojie Pan, Yuan Liu, Kehan Chen, Minghui Qiu, Wei Zhou, Jun Huang, Haiqing Chen, Wei Lin, and Deng Cai. 2021. Mell: Large-scale extensible user intent classification for dialogue systems with meta lifelong learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 3649–3659.

[28] Dian Yu, Luheng He, Yuan Zhang, Xinya Du, Panupong Pasupat, and Qi Li. 2021. Few-shot Intent Classification and Slot Filling with Retrieved Examples. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 734–749. doi:10.18653/v1/2021.naacl-main.59

[29] Dian Yu, Luheng He, Yuan Zhang, Xinya Du, Panupong Pasupat, and Qi Li. 2021. Few-shot intent classification and slot filling with retrieved examples. *arXiv preprint arXiv:2104.05763* (2021).

[30] Yunhua Zhou, Jiawei Hong, and Xipeng Qiu. 2023. Towards Open Environment Intent Prediction. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 2226–2240. doi:10.18653/v1/2023.findings-acl.140

[31] Yingying Zhuang, Yichao Lu, and Simi Wang. 2021. Weakly Supervised Extractive Summarization with Attention. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Haizhou Li, Gina-Anne Levow, Zhou Yu, Chitralekha Gupta, Berrak Sisman, Siqi Cai, David Vandyke, Nina Dethlefs, Yan Wu, and Junyi Jessy Li (Eds.). Association for Computational Linguistics, Singapore and Online, 520–529. doi:10.18653/v1/2021.sigdial-1.54

[32] Yingying Zhuang, Jiecheng Song, Narayanan Sadagopan, and Anurag Beniwal. 2023. Self-supervised Pre-training and Semi-supervised Learning for Extractive Dialog Summarization. In *Companion Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) *(WWW '23 Companion)*. Association for Computing Machinery, New York, NY, USA, 1069–1076. doi:10.1145/3543873.3587680

Appendix

A Data Anonymization

Due to business considerations, we are not permitted to share the results using the original customer data. As a result, we manually anonymized both the labels and transcripts to ensure no personal information is included. Additionally, specific product and service names were denonymized to prevent the identification of the company from the transcript or label descriptions. Despite these modifications, the conclusions drawn from our experiments remain valid.

B Impact of Retrievers

In order to evaluate the impact of retrievers on the final performance, we experimented four different retrievers in REIC including one sparse retrieval method and three dense retrievers, details in Appendix D. The intent detection accuracy across different business verticals using these retrievers is presented in Table 3.

BM25, despite being an unsupervised sparse retrieval method, performs competitively, achieving an overall accuracy of 0.532. Among the dense retrievers, MPNet outperforms the others, attaining the highest accuracy across both the 3P Business and 1P Business verticals. This suggests that MPNet's contrastive learning-based sentence embeddings are highly effective for retrieving relevant candidates that aid intent classification. In contrast, Contriever exhibits the lowest accuracy across all categories.

Our findings show that retriever selection significantly impacts intent classification. Although BM25 is a strong baseline, dense retrievers like MPNet consistently outperform it. This highlights the value of high-quality embeddings and extensive fine-tuning on large datasets, which is why we have chosen MPNet as our final retriever in REIC. We also conducted experiments to study the impact of retrieval candidate size in Appendix C.

Table 3: Intent detection accuracy on different business verticals using different retrievers in REIC

Retriever	3P Business	1P Business	Overall
BM25	0.521	0.537	0.532
Contriever	0.450	0.461	0.457
ColBERTv2	0.503	0.560	0.542
MPNet	**0.545**	**0.573**	**0.564**

C Impact of Retrieval Candidate Size

We investigated the impact of different retrieval candidate numbers (top-k) in REIC to balance intent detection accuracy and inference latency. The Figure 3 illustrates the trade-off between these two factors, with overall accuracy plotted on the left y-axis (blue) and inference latency on the right y-axis (red) against different values of top-k. From the accuracy perspective, increasing top-k allows the model to access a broader range of relevant information, leading to better predictions. Beyond a certain threshold, additional retrieved candidates contribute minimally to accuracy while still increasing computational complexity. Latency, on the other hand, exhibits a sharp rise as top-k increases. This indicates a crucial trade-off: although retrieving more candidates can improve accuracy, it also leads to longer inference times, which may not be suitable for real-time applications.

In our experiments, we select top-k = 10 which ensures a meaningful accuracy boost without incurring excessive computational costs. However, the ideal top-k may vary depending on application requirements. For instance, real-time systems such as customer service chatbots or voice assistants may favor a lower top-k to maintain fast response times. Conversely, offline or batch-processing applications could accommodate higher top-k values if maximizing accuracy is a priority. Our findings emphasize the need to carefully tune retrieval parameters in REIC to meet specific operational demands.

D Implementation Details

To fine-tune our REIC LLM, we apply 8 NVIDIA-A100 40GB GPUs with 96 vCPUs to conduct PEFT [13] training with LoRA adapters [8]. We choose a set of LoRA parameters with a rank of 8, an alpha value of 16, and a dropout rate of 0.1. The training batch size is set to 8 per GPU with a learning rate of $2e^{-5}$. We train the model using Cross Entropy Loss for 3 epochs which takes around 3 hours on the instance.

We experimented with the following off-the-shell retrievers for candidate retrieval:

- **BM25** [19] is a widely used traditional sparse retrieval method. Although it is unsupervised, it consistently demonstrates strong performance across a variety of benchmarks[26].
- **MPNet**[4] [25] is a sentence embedding model fine-tuned on one billion sentence pairs using a contrastive learning objective.

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

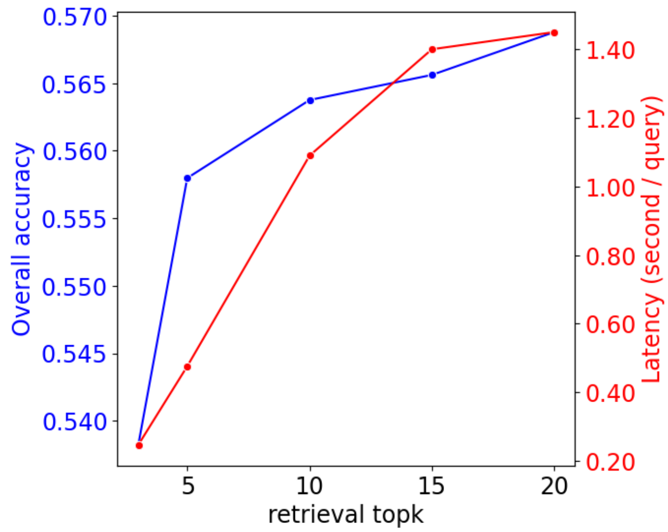

Figure 3: The accuracy and latency when using different retrieval top-k values.

- **Contriever-MS MACRO** [9] is an unsupervised dense retriever pre-trained with contrastive learning and fine-tuned on MS MARCO [15].
- **ColBERT-v2** [20] is a late-interaction retriever that combines denoised supervision and residual compression to improve retrieval quality and reduce space footprint.