# Building Multi-turn Intent Classification with LLM-based Labeling

**Anonymous ACL submission**

## Abstract

Intent classification is essential for customer service routing, connecting customers to the appropriate agents and reducing handling time and operational cost. Developing a real-world multi-turn intent classification system is challenging due to complex intent taxonomies, dynamic intent switching within conversations, and limited labeled training data. To address these challenges, we propose a scalable multi-turn intent classification framework for e-commerce customer service that models intent along multiple dimensions. We introduce LLM-based labeling strategies to annotate real customer transcripts at scale and augment training with LLM-simulated multi-turn dialogues that expand coverage of topic and intent switches, which are rare in existing transcripts. Through extensive experiments, we find that explanation-guided labeling with a self-critique step produces the most accurate training labels. Fine-tuned models built on a RoBERTa backbone outperform zero-shot LLM prompting while achieving substantially lower inference latency. Finally, we show that a hybrid approach that combines the fine-tuned classifier with LLM prompting further improves accuracy over either component alone. Overall, our results provide practical guidance for building and deploying high-accuracy, low-latency, large-scale multi-turn intent classification systems.

## 1 Introduction

Agentic customer service has become increasingly important for e-commerce platforms (Cui et al., 2017; Zhou et al., 2023). Different agents, either LLMs (Large Language Models) or workflows, are designed or trained to handle specific customer issues. Intent detection is therefore crucial for efficient routing. Intent classification failures may lead to irrelevant responses or unnecessarily escalate to human agents, increasing operational cost and degrading customer experience (Qi et al., 2021).

In recent years, LLMs have demonstrated strong potential for improving intent detection due to their few-shot generalization and broad world knowledge (Zhao et al., 2023; Arora et al., 2024). However, deploying intent models that achieve both high accuracy and low latency at scale remains challenging in customer service.

First, **scalability** becomes a bottleneck as business domains or product lines expand.In real-world e-commerce systems, a high-level intent such as *"Cancel"* or *"Refund"* may be associated with multiple products. Modeling intents at a product-specific level leads to label explosion and increasing model complexity. Moreover, maintaining a consistent intent taxonomy and obtaining sufficient labeled training data becomes difficult at scale (Qi et al., 2021; Liu et al., 2024a). Additional **scalability** challenges arise from compound intents. User utterances may express multiple, non-mutually exclusive intents, such as requesting an order cancellation while reporting an unrecognized charge. Traditional flat intent classifiers are ill-suited to this.

Second, **context carryover and intent switching** increase the difficulty of intent detection in multi-turn conversations. The domain of user intent might be inferred from prior context, while user goals can shift in mid-dialogue. For example, a customer may initially seek help troubleshooting a service issue and later shift to requesting service cancellation. Without modeling conversational history, a system may misattribute the current intent or fail to link it to the relevant context. Prior work has incorporated contextual signals for intent prediction(Wu et al., 2021; Nandi et al., 2024), but typically assumes access to comprehensive labeled multi-turn data or context features. Addressing topic shifts and intent switches in real customer-service applications remains underexplored.

Finally, **low-latency requirements** constrain practical deployment in production. Although LLMs

have strong potential for intent classification, high latency makes them unsuitable for real-time inference. Production deployments must therefore balance classification accuracy with computational efficiency (Liu et al., 2024a).

To address these challenges, we (i) decompose intent understanding into three dimensions—*Domain*, *Intent*, and *Issue*—so that adding new domains or intents expands only the relevant dimension. This reduces the effective label space and supports compound intent modeling by separating actionable intent types (e.g., *"Cancel"*, *"Informational Inquiry"*) from issue attributes (e.g., *"Unrecognized Charge"*, *"Sync/Download"*). We then (ii) fine-tune lightweight models leveraging various LLMs-based methods to generate labels from customer agent transcripts at scale. The model processes the concatenation of the dialogue history and the current turn, enabling context-aware, turn-level intent detection. Finally, (iii) we augment training data with LLM-simulated multi-turn dialogues that inject topic and intent switches—patterns that are rare in transcripts but critical for robustness.

## 2 Related work

### 2.1 Multi-turn intent classification

Multi-turn intent classification incorporates dialogue context to improve intent prediction. Prior work uses contextual encoders for customer service intent detection and models cross-turn dependencies via hierarchical or graph-based structures (Wang et al., 2021; Senese et al., 2020; Liu and Chen, 2019; Qin et al., 2021). We follow this line but focus on data-driven robustness to context carryover and intent switching via controllable simulation.

LLMs have also been explored for intent detection through prompting and hybrid routing (Arora et al., 2024), as well as retrieval-augmented or demonstration-based pipelines for few-shot intent prediction (Yu et al., 2021; Zhang et al., 2025; Liu et al., 2024b). In contrast, we primarily use LLMs for automatic labeling and data generation to train lightweight models, with optional LLM fallback at inference time.

### 2.2 User simulation

User simulation is widely used for synthetic dialogue generation and system evaluation, including agenda-based and neural approaches (Schatzmann et al., 2007; Lin et al., 2021; Sun et al., 2022). Recent work shows LLMs can act as user simulators (Balog and Zhai, 2025; Balog et al., 2025). Unlike simulators aimed at general task completion, our simulator targets real-world multi-turn behaviors—especially intent switching—using control to maintain coherence and constrain generation.

### 2.3 Chain-of-Thought and self-critique reasoning

Chain-of-Thought and self-refinement improve LLM outputs via intermediate reasoning and iterative revision (Wei et al., 2022; Kojima et al., 2022; Madaan et al., 2023; Shinn et al., 2023). We apply these ideas to transcript labeling (rather than inference-time in-context learning) to generate higher-quality training data for low-latency intent models. Compact transformer students further support efficient deployment (Sanh et al., 2019; Jiao et al., 2020; Wang et al., 2020).

## 3 Preliminary

We formulate multi-turn intent classification as a multi-class text classification problem. Given a conversation history $\mathcal{C}$ and the customer's current utterance $q$, the objective is to predict an intent label $t$ from a predefined set $T = \{t_1, \ldots, t_k\}$ using a model $\mathcal{M}$. The predicted intent $\hat{t}$ is obtained by maximizing the posterior probability:

$$\hat{t} = \arg\max_{t \in T} P(t \mid q, \mathcal{C}; \theta), \qquad (1)$$

where $P(t \mid q, \mathcal{C}; \theta)$ denotes the probability of intent $t$ conditioned on $(q, \mathcal{C})$, parameterized by $\theta$ of $\mathcal{M}$. As mentioned in Section 1, incorporating both $q$ and $\mathcal{C}$ in our model is essential because of context carryover and intent switching in multi-turn settings.

The number of intents $k$ grow rapidly as product lines expand. To address the scalability issue, we propose an ontology that captures complementary aspects of intent understanding (Figure 1): **Domain**, representing product categories; **Intent**, distinguishing conversational intents from actionable intents; and **Issue**, representing slot-level attributes associated with each intent. This decomposition reduces classification complexity as well as supports **compound intent modeling** naturally by decoupling intent identification from issue detection, allowing the system to jointly predict an intent and its associated issue.

We train separate models to classify *Domain*, *Intent*, and *Issue*. Their outputs are then com-

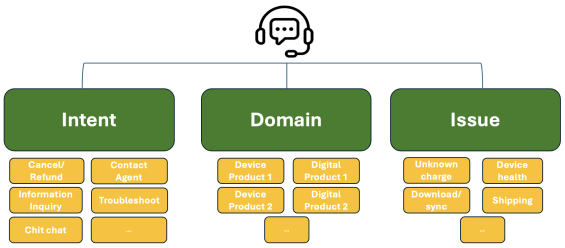


Figure 1: Intent classification ontology.

posed and mapped to downstream actions, including workflows, LLM-based agents, or human-agent hand-offs (Figure 2). Considering the latency requirement, we leverage a foundation model, Claude 3.7 Sonnet[1], to generate high-quality labeled training data. In the following sections, we introduce and evaluate multiple LLM-based labeling and data augmentation strategies that leverage existing transcript data to finetune our models under this ontology.

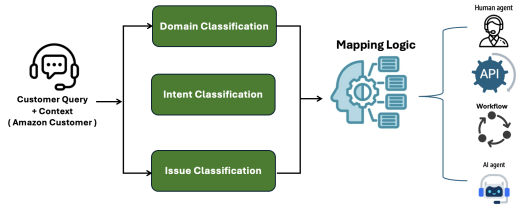


Figure 2: Intent detection system architecture

## 4 Method

### 4.1 Labeling strategy for intent classification

In real conversations, either the agent or the customer may consecutively input several utterances. However, the dialogue between a customer and a chatbot is typically conducted in an alternating manner. To construct data that aligns with the chatbot format, we merge consecutive utterances from the same role into a single unit, resulting in a dialogue $d = [q_1, a_1, ..., q_n, a_n]$ where $q_i$ represents the user's current query and $a_i$ represents the agent's response. For a query $q_i$, we define its context or conversation history as $C_i = [q_1, a_1, ..., q_{i-1}, a_{i-1}]$. In the following, we propose different label generation strategies for building our intent classification models.

**Single-stage reasoning-guided labeling**

We prompt the LLM to infer the customer's intent given the current user query $q_i$ and its associated conversation history $C_i$. The prompt provides the taxonomy along with descriptions of each label, and instructs the LLM to output a single label accompanied by a free-form explanation. This explanation cites the key evidence in both the current query $q_i$ and the context $C_i$ that supports the model's decision. We hypothesize that asking the LLM to articulate its reasoning and supporting evidence leads to more accurate intent classification by itself.

**Two-stage labeling with self-critique**

We adopt a two-stage labeling strategy that decomposes intent annotation into an *initial prediction* followed by an explicit *self-critique and revision*. **Stage 1** applies the single-stage reasoning-guided labeling procedure described in the previous section, producing an intent label with a brief explanation. In **Stage 2**, the LLM is given the original input together with the Stage-1 prediction and rationale, and is prompted to act as a critic. It assesses consistency with the dialogue evidence and intent taxonomy in Stage-1, flags failures (e.g., reliance on spurious keywords, missed contextual signals in $C_i$, confusion between closely related intents, or hallucinated assumptions), and then either confirms the Stage-1 label or revises it with a short justification. Stage 2 must explicitly indicate whether the label is *kept* or *revised*; when revised, it must cite minimal supporting evidence (e.g., a specific utterance in $C_i$ or phrase in $q_i$) motivating the correction. We hypothesize that this two-stage self-critique improves accuracy by correcting errors introduced during initial reasoning.

To evaluate our hypothesis, we manually reviewed more than 5,500 randomly sampled test instances and report the results in Table 1. The single-stage and two-stage strategies produced identical annotations for approximately 87% of samples, and these were largely accurate. In the remaining 13% of

[1] https://www.anthropic.com/news/claude-3-7-sonnet

cases where the strategies disagreed, the two-stage approach was correct in the majority of instances, suggesting that the second-stage reassessment reduces hallucinations and overall annotation errors.

### 4.2 Data augmentation strategy for intent classification

Although our e-commerce platform provides access to millions of real customer service transcripts between agent and customers that can be leveraged to train models, these transcripts are typically single-topic and follow a largely linear progression. In contrast, customer–bot interactions are potentially dynamic: users may switch intents or change goals in a single session. Consequently, models trained solely on real transcripts often fail to generalize to these complex patterns.

**Data generation using dialogue simulator:** To bridge this gap, we develop a multi-turn dialogue generation framework that simulates realistic customer–bot interactions. Our framework uses an LLM-based user query simulator that interacts with an LLM-based response simulator, while a *Simulator Controller* orchestrates the conversation by initializing dialogues, managing turn-by-turn flow, and introducing topic/intent shifts when appropriate. The controller combines intent-level planning with controlled randomness to elicit underrepresented behaviors in real transcripts (e.g., chit-chat, intent changes, follow-up and clarification questions). At each turn, it selects the next user intent and prompts the user simulator to produce the corresponding utterance, after which the response simulator generates the bot reply. We also inject alternative seed queries sampled from a multi-domain top-query database to facilitate topic and intent transitions. The overall interaction flow is shown in Figure 3 and simulation prompt template can be found in Appendix C.3. Finally, we use the two stage labeling strategy in Section 4.1 to automatically label simulated dialogues.

### 4.3 Hybrid approach of LLM and fine-tuned model

To demonstrate how a fine-tuned lightweight model can be combined with an LLM to balance accuracy and latency, we introduce a *hybrid intent detection strategy* that couples fast local classification with selective LLM escalation. As shown in Appendix C.5 and Figure 4, the lightweight model first outputs an intent probability distribution along with a confidence score. If the confidence score exceeds

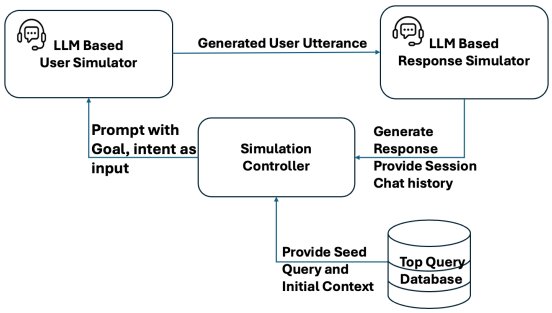


Figure 3: Multi-turn conversation simulator

a threshold $\tau$, we accept the lightweight model's prediction. Otherwise, we extract the model's top $K$ intent candidates and invoke an LLM to perform *constrained* disambiguation over this candidate set. This hybrid design routes most intent detection requests through the low-latency lightweight model, while reserving LLM calls for a small subset of ambiguous cases. The choice of $\tau$ and $K$ is motivated by the accuracy vs coverage analyses in Appendix C.6 (Figures 5 and 6).

## 5 Experimental setup

### 5.1 Datasets

We apply the anonymization procedure described in Appendix A. Each sample consists of conversation history, the current customer utterance, and its corresponding annotations for intent, issue, and product category. To build a balanced dataset, we use stratified sampling across months and domains. Since some categories are significantly overrepresented, we further downsample high-frequency categories when forming the final training and evaluation splits. We provide additional dataset construction details are provided in Appendix B.

### 5.2 Evaluation benchmarks

We consider the following baselines to evaluate and benchmark our LLM-based labeling and data augmentation methods:

- **RoBERTa**: We fine-tune RoBERTa-base (Liu et al., 2019) with multi-head classification components (*Intent*, *Domain*, and *Issue*), enabling simultaneous prediction across multiple label spaces. To compare prompting and data-generation choices for fine-tuning (Section 4), we train variants using: (i) single-stage reasoning-guided labeling, (ii) two-stage labeling with self-critique, and (iii) two-stage labeling with

**Table 1a: Stage wise metric breakdown**

| Outcome | Count |
|---|---|
| Stage 2 is correct | 566 |
| Stage 1 is correct | 84 |
| Both Stage 1 & Stage 2 are incorrect | 70 |
| Both Stage 1 & Stage 2 are correct | 4802 |
| Total | 5522 |

**Table 1b: Overall accuracy**

| Outcome | Accuracy |
|---|---|
| Stage 2 is correct | 97% |
| Stage 1 is correct | 88% |

Table 1: Stage 1 and Stage 2 performance analysis

self-critique plus dialogue-simulator augmentation. We provide implementation details in Appendix C.4.

- **Claude Sonnet 3.7 zero-shot**: We evaluate Claude Sonnet 3.7 in a zero-shot setting using a prompt that enumerates the intent taxonomy and provides brief definitions for each label.
- **Nova Pro zero-shot**: We mirror the **Claude Sonnet 3.7 Zero-shot** setup but replace Claude with Nova Pro, a smaller LLM that offers lower inference latency.
- **Hybrid approach (fine-tuned RoBERTa + Claude Sonnet 3.7)**: Following Section 4.3, we first use the best-performing fine-tuned RoBERTa model to retrieve the top-3 candidate labels for each classifier (*Intent*, *Domain*, *Issue*). We then prompt Claude Sonnet 3.7 to select the final label conditioned on these candidates. This hybrid baseline tests whether zero-shot prompting can further improve over RoBERTa fine-tuning.

# 6 Results

## 6.1 Automatic evaluation on fine-tuned RoBERTa models

As described in Sections 4.1 and 4.2, we employ multiple data labeling and augmentation strategies to generate training data for fine-tuning RoBERTa models. Specifically, we consider three approaches for comparison: (1) a single-stage prompting strategy, (2) a two-stage prompting strategy, and (3) a two-stage prompting strategy + simulated dialogue data augmentation. We generate ground-truth labels using an LLM according to each strategy. We report automatic evaluation results in Table 2. Models trained under all three strategies achieve comparable performance across classification tasks, indicating that the fine-tuned models are able to effectively learn from their corresponding LLM teacher models. Notably, models fine-tuned using the two-stage prompting strategy exhibit substantial performance gains on the issue classification task. We hypothesize that issue classification involves greater ambiguity and is therefore more challenging. The two-stage prompting strategy, which encourages additional self-verification and refinement, produces higher-quality labels and improves fine-tuning performance.

| Model Type | Single-Stage Labeling | Two-Stage Labeling | Two-Stage Labeling + Data Augmentation |
|---|---|---|---|
| Intent Model | 80.70% | 80.30% | 79.50% |
| Product Model | 80.90% | 80.80% | 81.70% |
| Issue Model | 76.70% | 79.10% | 78.20% |

Table 2: Automated evaluation accuracy results

## 6.2 Human evaluation

To comprehensively evaluate the effectiveness of our labeling and data augmentation strategies, as well as the benefits of the hybrid approach, we compare across several baseline approaches (see Section 5.2) using a 1,206 human-annotated dataset. The dataset is constructed by random sampling, with 50% from the evaluation dataset described in Section 5.1 and the remaining 50% from production traffic. The results, presented in Table 3, illustrate performance across all 3 types of classification models and an overall aggregate assessment based on Precision, Recall, and F1-score.

Overall, our results show that models fine-tuned with the two-stage labeling strategy and augmented with simulated dialogue data outperform alternative approaches. While training with data generated via single-stage prompting already yields strong performance, adding self-critique stage further improves labeling accuracy and model performance. Moreover, incorporating simulated multi-turn dialogues helps the model better handle intent switches, contributing to additional gains (F1 score increases by 2%-5% compared to single-stage prompting). Finally, we observe that intent classification perfor-

| | Intent Model | | | Product Model | | | Issue Model | | |
|---|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | F1 Score | Precision | Recall | F1 Score | Precision | Recall | F1 Score |
| RoBERTa(Single-stage labeling) | 73% | 71% | 72% | 73% | 65% | 69% | 67% | 61% | 64% |
| RoBERTa(Two-stage labeling) | 74% | 73% | 73% | 72% | 69% | 70% | 72% | 64% | 68% |
| RoBERTa(Two-stage labeling+data augmentation) | 75% | 74% | 74% | 75% | 72% | 73% | 73% | 66% | 69% |
| Hybrid approach | 77% | 75% | 76% | 78% | 74% | 76% | 74% | 68% | 71% |
| Nova Pro zero-Shot | 48% | 42% | 45% | 61% | 52% | 56% | 42% | 35% | 38% |
| Claude Sonnet 3.7 zero-shot | 75% | 67% | 71% | 61% | 69% | 65% | 54% | 51% | 52% |

Table 3: Performance comparison across different baselines for intent, product, and issue classification

mance is further improved by a hybrid approach that combines a fine-tuned RoBERTa model with LLM-based prompting.

Purely prompting-based methods (e.g., Claude zero-shot and Nova zero-shot) exhibit inferior performance. This degradation likely stems from the absence of domain-specific customer service training data, which increases the likelihood of hallucinations and intent misclassification. Although the hybrid approach also employs prompting when model confidence is low, it substantially outperforms standalone prompting. This gain stems from using the fine-tuned RoBERTa model to first retrieve a limited set of relevant intent candidates, thereby constraining the LLM's decision space and enabling more accurate final predictions.

We also evaluate P50 and P90 latency across different methods, with results summarized in Table 4. The findings indicate that RoBERTa achieves the lowest latency, approximately 20–50× faster than LLM zero-shot. While the hybrid approach delivers improved accuracy, it incurs higher latency than RoBERTa, suggesting a trade-off between performance gains and inference efficiency.

| Model | P50 Latency | P90 Latency |
|---|---|---|
| RoBERTa | 0.08s | 0.1s |
| Nova Pro Zero-shot | 1.98s | 3.97s |
| Claude Sonnet 3.7 Zero-Shot | 5.93s | 7.52s |
| Hybrid (RoBERTa + Zero-shot) | 1.99s | 3.1s |

Table 4: Model latency performance comparison

### 6.3 Online deployment performance

We deployed the intent classification model (fine-tuned RoBERTa-base) in an e-commerce customer service production systems. Compared to the previously deployed single-turn intent detection system, which could not support topic shifts or context carryover in an ongoing sessions, our model enables seamless and dynamic intent routing in multi-turn interactions. In a one-month online A/B test, our model increased the bot automation rate by 4.91% and improved the positive customer response rate by 7.89%, demonstrating benefits for both customer experience and operational efficiency while achieving low end-to-end production latency (P50: 0.12 s; P90: 0.16 s; P99: 0.20 s).

## 7 Conclusion

In this paper, we address multi-turn intent detection for customer service applications. To handle the scalability and heterogeneity of intent taxonomies, we propose an ontology that captures complementary facets of user intent. To mitigate the scarcity of annotated data, we introduce LLM-based labeling methods that generate high-quality supervision from existing customer transcripts, and augment training with LLM-simulated multi-turn dialogues that explicitly model topic shifts and intent switches.

Our experiments show that two-stage labeling with self-critique, combined with simulated dialogue augmentation, consistently outperforms alternative labeling strategies. The resulting fine-tuned RoBERTa models outperform pre-trained LLMs in zero-shot settings while achieving substantially lower latency. A hybrid routing strategy that combines fine-tuned RoBERTa with an LLM further improves performance on ambiguous cases.

Our findings provide actionable guidance for practitioners building production multi-turn intent detection systems by effectively combining real transcript data with LLM-generated dialogues.

## 8 Limitations

Our evaluation is based on customer service conversations from a specific set of products, locales, and workflow designs, so results may not fully generalize to other domains with different intent taxonomies or dialogue patterns. In addition, model quality depends on the consistency of upstream labels (human or LLM-assisted); any ambiguity in intent definitions or noise in automatic labeling can

propagate to training and inflate offline estimates. Finally, offline metrics may not perfectly translate to end-to-end customer impact because real deployments involve additional constraints (policy, UI, latency, and fallback behavior) and are subject to distribution shift over time; broader cross-domain/locale testing and controlled online studies are needed to validate robustness and user outcomes.

## References


Gaurav Arora, Shreya Jain, and Srujana Merugu. 2024. Intent detection in the age of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*.

Krisztian Balog, Nolwenn Bernard, Saber Zerhoudi, and ChengXiang Zhai. 2025. Theory and toolkits for user simulation in the era of generative AI: User modeling, synthetic data generation, and system evaluation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*, pages 4138–4141, Padua, Italy. Association for Computing Machinery.

Krisztian Balog and ChengXiang Zhai. 2025. User simulation in the era of generative AI: User modeling, synthetic data generation, and system evaluation. *arXiv preprint arXiv:2501.04410*.

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*, pages 97–102.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Hsien-chin Lin, Nurul Lubis, Songbo Hu, Carel van Niekerk, Christian Geishauser, Michael Heck, Shutong Feng, and Milica Gasic. 2021. Domain-independent user simulation with transformers for task-oriented dialogue systems. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 445–456, Singapore and Online. Association for Computational Linguistics.

Junhua Liu, Yong Keat Tan, Bin Fu, and Kwan Hui Lim. 2024a. Balancing accuracy and efficiency in multi-turn intent classification for LLM-powered dialog systems in production.

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*.

Zhengyuan Liu and Nancy Chen. 2019. Reading turn by turn: Hierarchical attention architecture for spoken dialogue comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Zihan Liu, Yiming Chen, Hao Zhang, et al. 2024b. Lara: Linguistic-adaptive retrieval-augmentation for multi-turn intent classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Nanyun Raja, Shivang Gulati, Shubham Tan, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Subhadip Nandi, Neeraj Agrawal, Anshika Singh, and Priyanka Bhatt. 2024. Enhancing customer service chatbots with context-aware nlu through selective attention and multi-task learning. In *Proceedings of the 8th International Conference on Data Science and Management of Data (CODS-COMAD 2024)*, pages 220–228. Association for Computing Machinery.

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*, pages 304–310, Online. Association for Computational Linguistics.

Libo Qin, Zhou Chen, Wanxiang Che, Hang Li, and Ting Liu. 2021. Knowing where to leverage: Context-aware graph convolutional network with an adaptive fusion layer for contextual spoken language understanding.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York. Association for Computational Linguistics.

Matteo Antonio Senese, Giuseppe Rizzo, Mauro Dragoni, and Maurizio Morisio. 2020. MTSI-BERT: A session-aware knowledge-based conversational agent. In *Proceedings of the Twelfth Language Resources*

*and Evaluation Conference*, pages 717–725, Marseille, France. European Language Resources Association.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*.

Weiwei Sun, Shuyu Guo, Shuo Zhang, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2022. Metaphorical user simulators for evaluating task-oriented dialogue systems. *arXiv preprint arXiv:2204.00763*.

Peiyao Wang, Joyce Fang, and Julia Reinspach. 2021. CS-BERT: a pretrained model for customer service dialogues. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 130–142, Online. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Ting-Wei Wu, Ruolin Su, and Biing-Hwang Juang. 2021. A context-aware hierarchical BERT fusion network for multi-turn dialog act detection. In *Proceedings of Interspeech 2021*, pages 1239–1243.

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*, pages 734–749, Online. Association for Computational Linguistics.

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 the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1072–1080, Suzhou (China). Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Yunhua Zhou, Jiawei Hong, and Xipeng Qiu. 2023. Towards open environment intent prediction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2226–2240, Toronto, Canada. Association for Computational Linguistics.

# Appendix

## A Data anonymization

Due to business considerations, 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 Data construction

The dataset comprises both real customer service conversations and synthetic queries collected between 2024 and 2025. We apply a stratified sampling strategy across months and domains as follows: (i) to mitigate seasonality effects and prevent oversampling during peak periods, we sample an equal number of data points from each month. (ii) to reduce bias toward particular product lines or use cases, we further stratify the sampling process by selecting an equal number of examples per month across up to 30 existing skills(domains), where each skill corresponds to a specific task that an Amazon Customer Service representative can perform for a given product.

Because certain categories contain substantially more examples than others, we apply a downsampling procedure when constructing the final training and evaluation datasets. For each category, we cap the number of samples at 20,000 for training and 10,000 for testing. This downsampling strategy is applied consistently across all model types, including intent, domain, and issue classifiers. The test set was held out strictly for final evaluation.

## C Implementation details

### C.1 Classification prompt template

We use the following prompt template for conversation classification. It instructs the LLM to label each customer-service dialogue along three dimensions: conversational intent, issue type, and product category. For each example, the current utterance and its dialogue context are dynamically inserted into the template. The prompt provides detailed annotation guidelines, including (i) leveraging the full dialogue history for context-aware decisions and (ii) using structured reasoning to resolve ambiguous cases. We define three separate prompt templates corresponding to Stage 1, Stage 2 annotation and LLM zero-shot. Note LLM zero-shot does not ask LLM to provide justification explanation.

---

**Classification prompt template for stage 1**:

```
## Dialogue Context
{dialogue_context}

## Turn T - Current Customer Utterance
{turn_T_utterance}

## Conversational/Actional Intent Options (Choose Exactly One)

1. CANCEL - Request to cancel a subscription or service or product
2. REFUND - Request for a refund related to a service or subscription
3. CANCEL&REFUND - Explicitly state desire to cancel and request a refund
4. RETURN - Request for a return related to a service or product
5. REPLACEMENT - Request for a replacement related to a service or product
6. RETAIN/SUBSCRIBE - Request to retain/subscribe subscription and NOT to cancel
7. TRADE-IN - Customer is requesting to trade-in devices
8. ADS-FREE/ADS-REMOVAL - Customer is requesting to remove ads or subscribe ads free
9. TROUBLESHOOT - Customer describes an issue requiring troubleshooting
10. INFORMATIONAL_INQUIRY - General "how-to" or "what-is" questions
11. ISSUE_DEPENDENT_INQUIRY - Inquiries tied to a specific issue or service/product
12. Yes - The customer responded affirmatively
13. No - Customer generally respond with "No" or "Nope"
14. TRANSACTIONAL_OTHER - Other action-oriented or transactional inquiries
15. CHIT_CHAT/CONVERSATION_FILLER - Polite fillers, navigation phrases
16. FRUSTRATION/COMPLAINT - General dissatisfaction expressed
17. REQUEST_HUMAN_AGENT - Customer explicitly asks for a human agent
18. END_CONVERSATION - Customer clearly states issue is resolved
```

```
19. NON_AMAZON_TOPIC - Completely unrelated to Amazon products or services

## Issue Options (Choose Exactly One)

1. SETUP/REGISTRATION/ACTIVATION/INSTALLATION
2. CANCEL/REFUND/RETURN/REPLACEMENT/RETAIN/SUBSCRIBE/TRADE-IN/TRANSACTIONAL
3. CONNECT/PAIR
4. WIFI/NETWORK
5. SYNC/DOWNLOAD
6. PLAY/STREAMING/DISPLAY
7. AUDIO/SOUND
8. DAMAGE/REPAIR
9. WARRANTY
10. DEVICE_HEALTH
11. DEFECT
12. BATTERY
13. SOFTWARE/OTA UPDATE
14. LOST/STOLEN
15. PROMOTIONS/CREDITS/LOOT
16. RESTRICTIONS/PARENTAL_CONTROL
17. HOUSEHOLD
18. PAYMENT_ISSUE/PURCHASE/COINS
19. PRICING
20. UNRECOGNIZED_CHARGES/UNKNOWN_CHARGES/FRAUD_CHARGES
21. CONTENT_AVAILABILITY
22. SHIPPING/DELIVERY
23. OTHER_ISSUE
24. NONE

## Product Options (Choose Exactly One)

1.  Music
2.  Video
3. SmartTV

[... additional product categories ...]

## Output Format (JSON)
{
    "predicted_conversational_intent": "One of the 19 intent labels",
    "predicted_issue": "One of the 24 issue labels",
    "predicted_product": "One of the 31 product labels",
    "reason": "Brief explanation justifying the choice"
}
```

---

**Classification prompt for stage 2**:

```
## Taxonomy Intents

1. CANCEL - Request to cancel subscription/service/product or turn off
   auto renewal. Excludes refunds.
2. REFUND - Request for refund related to service/subscription.
3. CANCEL&REFUND - Explicitly cancel and request refund.
4. RETURN - Request for return related to service/product.
5. REPLACEMENT - Request for replacement related to service/product.
6. RETAIN/SUBSCRIBE - Request to retain/subscribe, NOT cancel.
7. TRADE-IN - Customer requesting to trade-in devices.
8. ADS-FREE/ADS-REMOVAL - Request to remove ads or subscribe ads free.
9. TROUBLESHOOT - Customer describes issue requiring troubleshooting.
10. INFORMATIONAL_INQUIRY - General "how-to" or "what-is" questions
    answerable from help pages. Excludes vague utterances like "I need
    help".
11. ISSUE_DEPENDENT_INQUIRY - Inquiries tied to specific issue/service/
    product requiring customer context. Exclude transactional intents.
12. Yes - Customer responded affirmatively. Don't confuse with
    END_CONVERSATION.
13. No - Customer responds "No" or "Nope". Don't confuse with
    END_CONVERSATION.
14. CHIT_CHAT/CONVERSATION_FILLER - Polite fillers (hi, thanks),
```

```
    navigation phrases. Exclude frustration/complaint and Yes/No.
15. FRUSTRATION/COMPLAINT - General dissatisfaction without clear request.
16. REQUEST_HUMAN_AGENT - Customer explicitly asks for human agent.
17. END_CONVERSATION - Customer states issue resolved. Don't confuse with
    polite chit-chat.
18. NON_RELATED_TOPIC - Completely unrelated to the e-commerce. Must be completely
    off-topic.

## Taxonomy Issues

1. SETUP/REGISTRATION/ACTIVATION/INSTALLATION - Issues registering,
   setting up, activating, or installing device/subscription/service/app.
2. CANCEL/REFUND/RETURN/REPLACEMENT/RETAIN/SUBSCRIBE/TRADE-IN/
   TRANSACTIONAL - Issue with transaction already occurred or attempted.
3. CONNECT/PAIR - Issues connecting or bluetooth pairing.
4. WIFI/NETWORK - Issues with networking or Wifi.
5. SYNC/DOWNLOAD - Trouble syncing/downloading (often eBook/App related).
6. PLAY/STREAMING/DISPLAY - Trouble playing content or visualizing.
7. AUDIO/SOUND - Audio issues (no audio, low audio, etc.).
8. DAMAGE/REPAIR - Device damage, inquiring about repair.
9. WARRANTY - Inquiring about or checking warranty.
10. DEVICE_HEALTH - Device performance/stability issues (crashes, reboots,
    bricked, frozen, responsiveness).
11. DEFECT - General mention of device defect.
12. BATTERY - Battery issues.
13. SOFTWARE/OTA UPDATE - Software update issues (stuck on update screen,
    no update available).
14. LOST/STOLEN - Reporting lost/stolen device.
15. PROMOTIONS/CREDITS/LOOT - Issues with promotions, bundles, credits,
    loot.
16. RESTRICTIONS/PARENTAL_CONTROL - Trouble with Parental Controls, Pins,
    child purchases.
17. HOUSEHOLD - Issues with household (sharing content, adding members).
    If PINs/Child Controls, select Parental Controls instead.
18. PAYMENT_ISSUE/PURCHASE/COINS - Issue/inquiry related to payment, gift
    cards, or coins.
19. PRICING - Price related issues (price match, price adjustment).
20. UNRECOGNIZED_CHARGES/UNKNOWN_CHARGES/FRAUD_CHARGES - Generic
    unrecognized charges customer not aware of or deems fraud.
21. CONTENT_AVAILABILITY - Issues with content availability.
22. SHIPPING/DELIVERY - Shipping or delivery related issues.
23. OTHER_ISSUE - Other action-oriented/transactional inquiries not
    mentioned above.
24. NONE - No specific issue, purely chit-chat, complaint or frustration.

## Taxonomy Products

1.  Music.
2.  Video
3.  Video Channels
4. SmartTV Cube
[... additional product categories ...]
```

---

### Classification prompt template for LLM zero-shot:

```
## Dialogue Context
{dialogue_context}

## Turn T - Current Customer Utterance
{turn_T_utterance}

## Conversational/Actional Intent Options (Choose Exactly One)

1. CANCEL - Request to cancel a subscription or service or product
2. REFUND - Request for a refund related to a service or subscription
3. CANCEL&REFUND - Explicitly state desire to cancel and request a refund
4. RETURN - Request for a return related to a service or product
5. REPLACEMENT - Request for a replacement related to a service or product
6. RETAIN/SUBSCRIBE - Request to retain/subscribe subscription and NOT to cancel
```

```
7. TRADE-IN - Customer is requesting to trade-in devices
8. ADS-FREE/ADS-REMOVAL - Customer is requesting to remove ads or subscribe ads free
9. TROUBLESHOOT - Customer describes an issue requiring troubleshooting
10. INFORMATIONAL_INQUIRY - General "how-to" or "what-is" questions
11. ISSUE_DEPENDENT_INQUIRY - Inquiries tied to a specific issue or service/product
12. Yes - The customer responded affirmatively
13. No - Customer generally respond with "No" or "Nope"
14. TRANSACTIONAL_OTHER - Other action-oriented or transactional inquiries
15. CHIT_CHAT/CONVERSATION_FILLER - Polite fillers, navigation phrases
16. FRUSTRATION/COMPLAINT - General dissatisfaction expressed
17. REQUEST_HUMAN_AGENT - Customer explicitly asks for a human agent
18. END_CONVERSATION - Customer clearly states issue is resolved
19. NON_AMAZON_TOPIC - Completely unrelated to Amazon products or services

## Issue Options (Choose Exactly One)

1. SETUP/REGISTRATION/ACTIVATION/INSTALLATION
2. CANCEL/REFUND/RETURN/REPLACEMENT/RETAIN/SUBSCRIBE/TRADE-IN/TRANSACTIONAL
3. CONNECT/PAIR
4. WIFI/NETWORK
5. SYNC/DOWNLOAD
6. PLAY/STREAMING/DISPLAY
7. AUDIO/SOUND
8. DAMAGE/REPAIR
9. WARRANTY
10. DEVICE_HEALTH
11. DEFECT
12. BATTERY
13. SOFTWARE/OTA UPDATE
14. LOST/STOLEN
15. PROMOTIONS/CREDITS/LOOT
16. RESTRICTIONS/PARENTAL_CONTROL
17. HOUSEHOLD
18. PAYMENT_ISSUE/PURCHASE/COINS
19. PRICING
20. UNRECOGNIZED_CHARGES/UNKNOWN_CHARGES/FRAUD_CHARGES
21. CONTENT_AVAILABILITY
22. SHIPPING/DELIVERY
23. OTHER_ISSUE
24. NONE

## Product Options (Choose Exactly One)

1.  Music
2.  Video
3. SmartTV

[... additional product categories ...]

## Output Format (JSON)
{
    "predicted_conversational_intent": "One of the 19 intent labels",
    "predicted_issue": "One of the 24 issue labels",
    "predicted_product": "One of the 31 product labels",
}
```

### C.2 System prompt

We define three separate prompt templates corresponding to Stage 1, Stage 2 annotation and LLM zero-shot. Note LLM zero-shot and Stage 1 annotation shares the same system prompt.

---

**Stage 1 or LLM zero-shot system prompt**:

```
You are an expert annotation assistant specializing in analyzing conversations
between customers and bots/agents. Your task is to classify each customer message
(Turn T) into its primary intent and the most relevant Amazon product or
service discussed.

Use the provided dialogue history for context, and ensure that classifications
adhere strictly to the predefined categories. Always select exactly one intent
```

```
and one product for each message, even if the product is inferred from the
context.

If the product is ambiguous but likely Amazon-related, choose 'Other'. If the
message is unrelated to Amazon, select 'NON_AMAZON_TOPIC' as the intent. Provide
clear reasoning for your classifications, referencing specific dialogue cues and
your decision-making process.
```

---

**Stage 2 system prompt**:

```
You are a strict annotation reviewer. Your job is to AUDIT a prior classification
(Stage 1) for a customer's Turn T, using the SAME taxonomy as Stage 1.
Goals:
- Verify that the predicted intent, issue, and product each match their definitions.
- Challenge the original reasoning (try to find contradictions or missing evidence).
- Correct any mistakes; otherwise confirm the original labels.
- Be conservative with ambiguous cases: only use NON_AMAZON_TOPIC when clearly
unrelated to Amazon; do not confuse CHIT_CHAT with END_CONVERSATION; do not confuse
YES/NO with functional intents.
- Always choose EXACTLY ONE intent, ONE issue, and ONE product from the Stage-1
taxonomy (no new labels).
- Keep reasoning concise and reference concrete spans from the dialogue
(short quotes). IMPORTANT: Do NOT repeat the Stage-1 JSON.
Produce the reviewer JSON ONLY using the final_* keys.
```

### C.3 Dynamic conversation simulation prompt templates

The following prompt templates were used in this study to serve a two-stage approach for generating realistic customer service conversations. The `system_prompt_dialogue_helper` and `user_intent_helper` work together to analyze existing conversation history and identify all possible customer intents (such as ChitChat, Frustration, IntentChange, or FollowUpQuestion) that could naturally occur next in the dialogue flow. Once potential intents are identified, the system randomly selects one and employs the `system_prompt_talk_to_bot` and `user_prompt_turn_helper` templates to generate authentic customer responses that align with the chosen intent. This dual-phase prompting strategy ensures that simulated conversations maintain conversational coherence while introducing realistic variability in customer behavior, enabling comprehensive testing of chatbot performance across diverse interaction scenarios.

---

**`system_prompt_dialogue_helper`**:

```
ROLE:

You are the dialogue helper for a user simulator helping find the intent for the
user given a conversation history.

TASK:
Select out all POSSIBLE intent from CANDIDATE LIST to carry on the conversation
given the previous Conversation history.

GUIDELINES:
1. Read through the whole conversation and identify the subset of
POSSIBLE INTENTS

CANDIDATE LIST:
1.ChitChat: Small talk loosely related to previous chat history
2.Frustration: Expression of frustration in the middle of conversation
3.intentChage: The user changes their request midway through a conversation, the
request can be related to previous Conversation history
4.FollowUpQuestion: The user asks follow-up questions that are related to previous
Conversation history
5.Clarification: The user asks for clarification for certain points in previous
Conversation history
6.Rambling: Speaker(s) ramble and repeat themselves. They may paraphrase themselves
7.ContactRealAgent: Request to speak to a real agent
8.EndConversation: End Conversation naturally when the issue or problem is resolved
```

Here is the input format
<Conversation history>
[provide chat_history here]
</Conversation history>

Here is the output format
<Possible Intents>
[provide possible intents here]
</Possible Intents>

---

**user_intent_helper**:

Here is your input:

<Conversation history>
{chat_history}
</Conversation history>

Now respond with what the customer would say next:

---

**system_prompt_talk_to_bot**:

ROLE:
You are a user engaging in a natural conversation with a customer service bot or agent. Your goal is to generate the next user turn based on the conversation history and the intent provided below.

OBJECTIVES
Conversation-Level Goal:
Seek a resolution (e.g., HOW-TO answer) to the seed query provided below.
Current Turn Goal:
Generate a user response that aligns with the current intent described below.

INTENT DEFINITIONS
ChitChat – Casual or light-hearted comments loosely related to the conversation.
Frustration – Expressions of annoyance or dissatisfaction.
IntentChange – The user changes their goal mid-conversation, potentially related to prior turns.
FollowUpQuestion – User asks a question that builds directly on prior discussion.
Clarification – User requests clarification about something mentioned previously.
Rambling – User paraphrases, repeats, or meanders while staying within the topic.
ContactRealAgent: Request to speak to a real agent
EndConversation: End Conversation naturally when the issue or problem is resolved

GUIDELINES:
-If no chat history exists, begin with the seed query.
-Respond naturally: ask relevant questions, express preferences, or make decisions as needed.
-If the bot successfully resolves the task and provides a reference number, reply only with: "I'm all set" (no additional text).
-If the bot is repetitive or unhelpful across multiple turns, escalate by using Contact Real Agent intent and say "talk to a real agent".
-Do not impersonate a bot or break character.
-Be concise and speak like a real customer in real life. Each response should be less than 25 words.
-If the intent is either ContactRealAgent or EndConversation. Be concise and the response should be straight foward.
Here is the input format
<seed_query>
[provide seed_query here]
</seed_query>
<Provided Intent>
[provide intent here]
</Provided Intent>
<Conversation history>
[provide chat_history here]
</Conversation history>
Here is the output format
<Current Turn>

```
[provide the current turn]
</Current Turn>
"""
```

---

**`user_prompt_turn_helper`**:

```
Here is your input:

<Conversation history>
{chat_history}

</Conversation history>

<Provided Intent>
{provided_intent}
</Provided Intent>

Now respond with what the customer would say next:
```

### C.4 RoBERTa model fine-tuning implementation details

We use RoBERTa-base for intent detection model, optimizing with cross-entropy and the Adam optimizer. Models are trained for 10 epochs on 8 NVIDIA A10 GPUs, with learning rates of 1e-5, batch sizes of 32 and early stopping(patience = 5) to prevent overfitting.

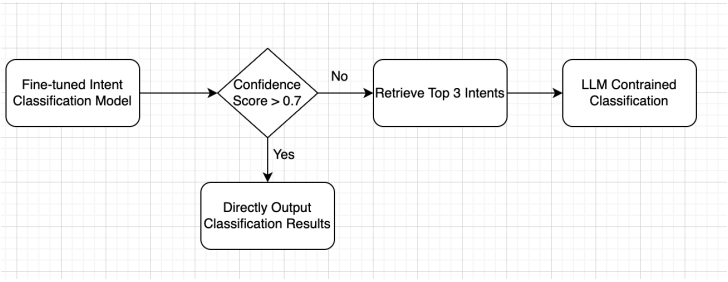


Figure 4: Hybrid deployment approach

### C.5 Hybrid approach work flow implementation Details

Figure 4 illustrates the workflow of our hybrid approach. In our implementation, we set the confidence threshold to $\tau = 0.7$ and use $K = 3$ candidate intents for LLM disambiguation (see Appendix C.6). We use a fine-tuned RoBERTa model as the lightweight classifier, and invoke Claude Sonnet 3.7 for intent classification when the confidence score falls below the threshold.

### C.6 Analysis of $\tau$ and $K$ in hybrid approach

We select the confidence threshold by balancing accuracy and coverage. As shown in Figure 5, setting the threshold to $0.7$ allows our fine-tuned model to cover roughly $80\%$ of intent-detection requests while maintaining about $85\%$ accuracy. We therefore choose $0.7$ as the operating point because it offers a practical tradeoff: the low-latency model can handle the majority of traffic with sufficiently high accuracy for direct deployment. For the remaining low-confidence cases (confidence $\leq 0.7$), we defer intent detection to an LLM. Importantly, for these instances, the ground-truth intent appears in our fine-tuned model's top-3 predictions nearly $90\%$ of the time (Figure 6). This suggests that low confidence typically reflects ambiguity among a small set of plausible intents rather than a complete failure. Accordingly, we ask the LLM to select among the top-3 candidate intents produced by the fine-tuned model.

This yields an effective hybrid intent detection strategy: high-confidence requests (confidence $\geq 0.7$) are handled directly by the lightweight model, while only a small fraction of ambiguous cases trigger a secondary LLM call over a constrained top-3 label set. This design improves accuracy on difficult utterances while keeping overall latency under control.

This is a section in the appendix.

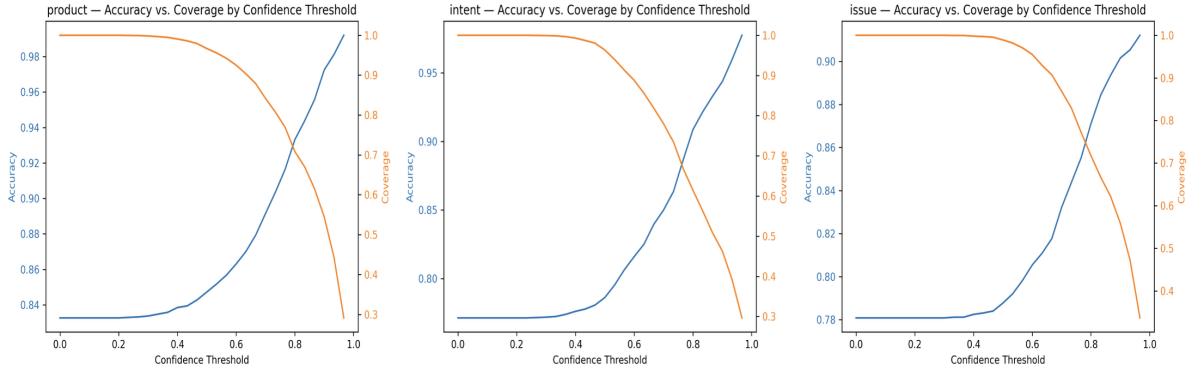


Figure 5: Accuracy–coverage tradeoff under different confidence thresholds

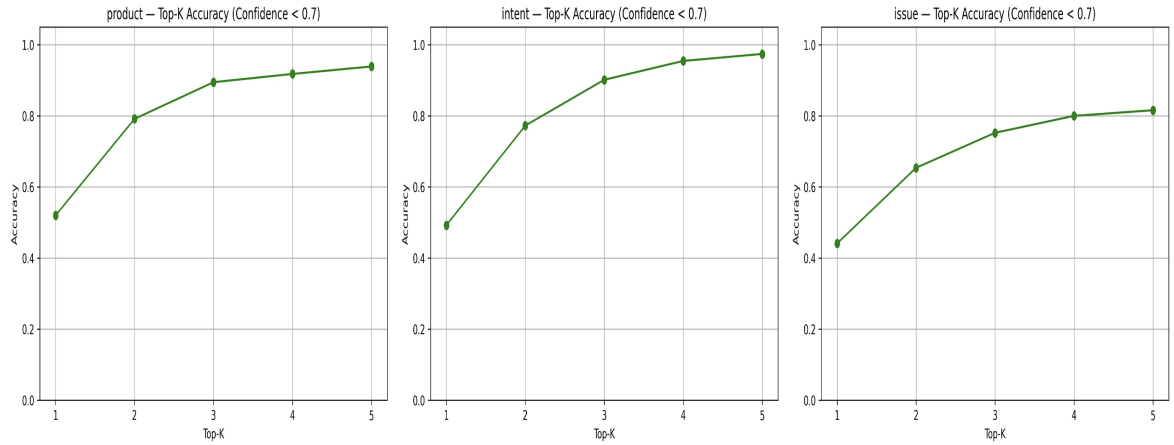


Figure 6: Top-$K$ accuracy for low-confidence predictions