Transfer Learning for Generalizable Automated LLM Improvement Pipeline for IVR Navigation

Vishal Sankar Ram, Jason Kushner, Manas Paldhe, and **Youngseo Son**
Infinitus Systems, Inc.
{vishal.ram, jason.kushner, manas.paldhe, youngseo.son}@infinitus.ai

Abstract

Administrative tasks in the healthcare domain share linguistic commonalities, but it can be time-consuming to manually design LLM prompts for each use case. When calling health insurers, interactive voice response (IVR) systems cause delays in patient care and increase provider burnout due to complex routing and long hold times. Thus, IVR navigation models can offer significant time savings and reduce barriers to care. We propose a production-quality automated LLM pipeline which leverages a small number of human-labeled ground truth datasets to transfer specialized prompts from one task to another; specifically, we perform a cross-task transfer of our IVR navigation logic, adapting the prompt from reaching the claims department to reaching the patient benefit department. Our approach reduces prompt complexity by up to 80% and obtains 82% turn-level accuracy in real-world industrial healthcare settings, surpassing a human-designed prompt at 79%.

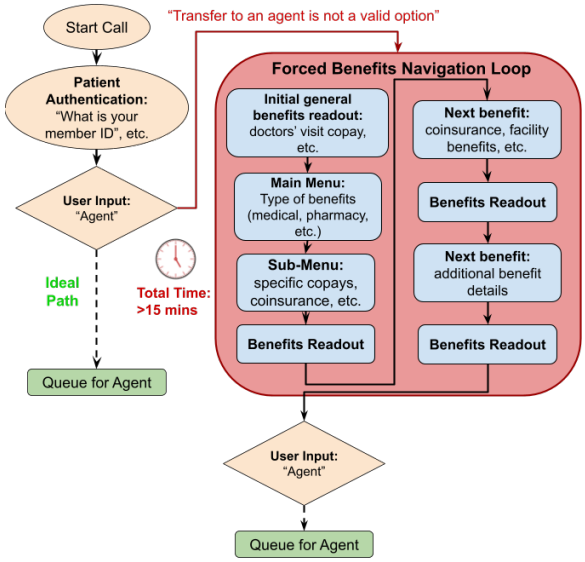

Figure 1: Example of a difficult IVR navigation where the system rejects a request for an agent and forces the user to listen to patient benefits before agreeing to a transfer.

1 Introduction

Administrative complexity in healthcare remains a critical bottleneck for patient care. Health insurers (payers) establish interactive voice response (IVR) systems as restrictive and difficult to navigate to save time for human agents. For prior authorization, 88% of physicians described authorizations as burdensome, with phone-based interactions prolonging the process (Hosfield, 2024).

Broadly, there are two types of IVR systems: 1) Traditional agents which follow a fixed set of logic and accept a narrow range of responses, and 2) modern agents which accept a larger variance in responses and may carry some context through the conversation (Rojas-Galeano, 2025). Agents designed to navigate these systems must account for both scenarios. Navigation for traditional IVR systems is challenging as menus are brittle - minor variations in the inputs can lead to unrecoverable states. Modern conversational agents offer improved flexibility but often introduce friction by enforcing strict requirements for agent escalations. A common example involves mandatory summaries of benefits, where the system requires the user to listen to specific plan information before a transfer request is possible (Figure 1). In extreme cases, users must navigate 15+ minutes of benefit summaries before reaching a human agent.

Reaching a human agent is often blocked by confusing IVR flows, rigid member-specific routing, and unreliable input recognition. In some cases, the only way to reach a live agent is to ignore instructions or enter incorrect information - for example, one insurer transfers calls to a human agent only if an incorrect date of birth is provided. For other payers, routing depends on the member ID: some callers can transfer to an agent, while others are limited to automated responses, voicemail callbacks,

or fax. It is also often unclear whether input failures are due to speech-recognition issues or incorrect information.

Deterministic IVR systems can provide a structured environment conducive to reinforcement learning, but achieving cross-payer generalization remains difficult with static reward models. Menus also change over time, necessitating sample-efficient adaptation both for training and inference. Additionally, once a navigation is learned, the model may struggle to apply it when trying to reach a different department within the same system. Hence, we define each navigation to a distinct department as a separate task. Transferring knowledge between tasks is difficult, as some responses, such as entering patient info, are shared between tasks, but others, such as structured menu responses, differ. Agent request timing and patient-specific internal routing can also differ for a single payer across tasks.

Recent transfer learning methods aim to normalize the input embedding space so that concepts are interpreted consistently across tasks (Zhao et al., 2023). In contrast, our approach operates at the prompt level: rather than modifying model weights or embeddings, we tune prompts to adapt behavior. This enables lightweight transfer across frontier models without requiring costly fine-tuning.

Contributions in this work include: **(1) Cluster-based prompt modularization**: An iterative system to cluster existing data and enable prompt optimization on a per-cluster basis. This reduces the complexity of LLM instructions by a large factor. **(2) Clustered Retrieval Augmented Generation (RAG)**: Dynamic instruction and hint injection improve performance without affecting optimization gains. **(3) Transfer learning:** A system to transfer implicit knowledge across analogous tasks and optimize prompts for each individual task.

2 Related Work

Deploying AI in healthcare presents a fundamental challenge: while the potential benefits are immense, the practical barriers to implementation remain formidable. Healthcare organizations face a plethora of challenges that make traditional AI deployment approaches prohibitively expensive and operationally infeasible.

The research community has developed sophisticated approaches to automated prompt optimization (Zhou et al., 2022), yet these methods face challenges for addressing healthcare's deployment constraints. Frameworks like DSPy (Khattab et al., 2023) and OPRO (Yang et al., 2024) have demonstrated impressive results in automating prompt engineering. DSPy treats prompting as an automated translation problem, where users describe tasks at a high level and the system generates optimized prompts using training examples. OPRO shows that LLMs can serve as meta-optimizers, iteratively refining prompts based on performance feedback. However, both approaches fundamentally require substantial task-specific training data to evaluate prompt candidates during optimization. In healthcare, where each payer-task combination is effectively a new use case, this data requirement makes independent optimization of every task data-intensive and time consuming (Wang et al., 2025).

Traditional transfer learning through fine-tuning, pre-training on large datasets, and adapting to specific tasks (Devlin et al., 2019; Brown et al., 2020) has been transformative in NLP. Even parameter-efficient alternatives like LoRA (Hu et al., 2022) reduce computational costs while maintaining performance. Yet fine-tuning remains impractical for many healthcare production environments without manual redaction of patient identifiers[1] in terms of both generalizability to new tasks and model maintenance.

Recent work has explored training single models across multiple related tasks, either through multi-task fine-tuning or unified prompt templates. However, research has revealed fundamental limitations to this approach. Studies on semantic diversity in multi-task training (Hong et al., 2025) show that when tasks require different reasoning strategies, performance actually degrades compared to task-specific models. In dialogue systems, researchers have found that designing specific, scoped tasks and building specialized models for each task outperforms monolithic approaches (Hakimov et al., 2024). For healthcare deployment, multi-task learning presents an additional operational challenge: tasks emerge sequentially over time as organizations expand AI adoption, but multi-task learning requires simultaneous access to all task data during training. This misalignment between research assumptions and production realities limits practical applicability.

An alternative approach leverages automated

[1]Healthcare entities and HIPAA requirements raise concerns regarding PHI being encoded indirectly through fine-tuning.

prompt optimization to enable transfer learning without parameter updates. Zhou et al. (Zhou et al., 2023) introduced APE, demonstrating that language models can iteratively refine prompts across tasks, though requiring substantial evaluation resources. Fernando et al. (2024) developed Promptbreeder for evolutionary prompt improvement, where evolved prompts could encode transferable reasoning patterns, though convergence can be slow. Most recently, Deng et al. (Deng et al., 2024) introduced PRewrite, using reinforcement learning to learn rewriting strategies that generalize across prompt types.

Despite these advances, the fundamental question remains: how do we effectively transfer knowledge across tasks? The key insight from recent work (Wang et al., 2023) is that transfer learning need not occur solely through model parameters. In-context learning demonstrates that LLMs can adapt to new tasks by transferring knowledge through carefully designed prompts rather than weight updates (Dong et al., 2024). Chain-of-thought prompting (Wei et al., 2022) shows that reasoning patterns the structure of how problems are decomposed, and transfer across tasks even when specific content differs.

This suggests an alternative path forward: rather than transferring learned parameters (via fine-tuning) or simultaneously optimizing multiple tasks (via multi-task learning), we can transfer optimized prompt structures from existing tasks to bootstrap new task prompts. Building on these insights, we introduce an automated transfer learning pipeline for prompt optimization designed for healthcare's constraints. Our method addresses use case heterogeneity through task clustering, creating specialized transferred prompts for related tasks (Shen et al., 2024) overcoming data scarcity by transferring optimized prompts with reducing expert involvement from weeks of manual design per task to validation and refinement. This approach can be improved even further with a reduced prompt complexity using semantic-based clustering and context-aware navigation using a dynamic RAG pipeline (Lewis et al., 2020)

Critically, our approach enables sequential task deployment: new tasks can leverage transferred prompts from previously optimized tasks without requiring simultaneous access to all task data or maintaining multiple model versions. This aligns with how healthcare organizations actually expand AI adoption, one payer at a time, one workflow at a time, while still preserving the benefits of cross-task knowledge.

3 Methods

We implement an end-to-end transfer learning pipeline which adapts prompts from one IVR navigation task type to another. In this paper, we used IVR navigation to the claims department as our source task and navigation to the patient benefit department as our target task. We designed a two-step pipeline: first, a transfer learning component adapts the prompt from our source task to our target task and optimizes prompts using the DSPy framework. Then, we extend this system by integrating clustering and retrieval-augmented generation (RAG) to improve robustness and scalability.

3.1 Automated LLM Prompt Transfer and Optimization

Transfer Learning Pipeline. The *Transfer Learning Pipeline* is composed of the *Prompt Transfer Pipeline* and the *Optimization Pipeline*. The *Prompt Transfer Pipeline* is an iterative module which takes information about source and target task data and returns a new prompt for the target task, then checks whether the prompt includes any hallucinated information. We use top-k prompts for each iteration and early stopping levers to prevent overfitting[2]. The *Optimization Pipeline* then uses selected DSPy optimizers to further increase the accuracy of the prompt.

Prompt Transfer Pipeline. This framework contains multiple LLM-based modules which automatically adapt prompts from the source task type to the target task type. Given a random target task example, the framework retrieves semantically similar examples from the datasets of both initial and target task via the *Source Task Example Retriever* and *Target Task Example Retriever* modules, both of which encode their respective datasets with text-embeddings models[3]. These examples are passed to the *Transfer Learning Hint Generator* module, which identifies the structural and action-level differences between the initial and target task. Both retrievers encode their datasets using 'text-embedding-004' from Vertex AI, selected for strong performance on short-text semantic similarity. Cosine similarity is used for retrieval with a

[2]See more details in Secion B.4.

[3]Find more details in Section A

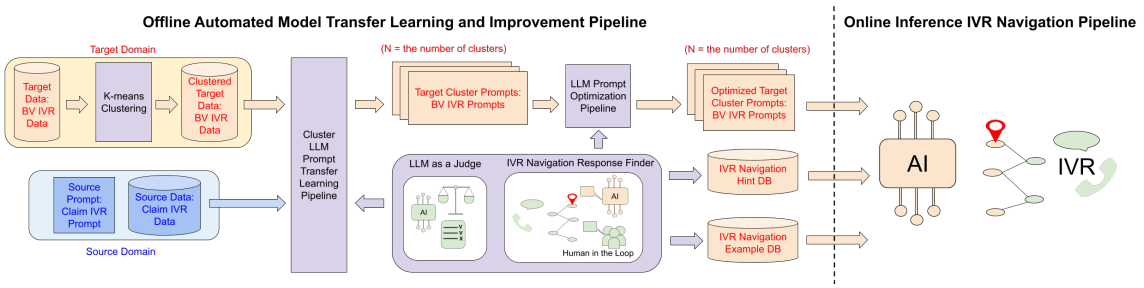

Figure 2: Automated LLM prompt task transfer and optimization pipeline. Claim IVR: IVR navigation for claims status follow up (source task), BV IVR: IVR navigation for patient benefit verification (target task).

top-k threshold[4] (see Figure 9).

Our *Prompt Generation Module* then uses task descriptions, sampled utterance-action pairs, an optimized source prompt, and the prior target prompt to generate a new prompt. To prevent the pipeline from becoming stuck in local minima, we designed an Aggressive LLM module *Prompt Similarity Aware Reducer* which compares the prompt from the previous iteration with the newly generated prompt, then aggressively reduces the token count and gives new syntax and structure to the prompt for high-similarity cases.

The resulting prompt is then verified by the *Prompt Hallucination Checker* module to detect hallucinated action names or invalid actions. If the prompt is hallucination-free, we evaluate the prompt on the training data of the target task. If there are hallucinations, the loop continues and uses the results generated from previous iterations. This allows the system to fall back and discard the current iteration if the prompt fails verification. Each component is implemented as a distinct DSPy signature so modules can be independently updated or replaced. During generation the pipeline propagates both the textual artifacts (prompts, hints) and structured diagnostics to enable interpretable adaptation and controlled fallback behavior.

Optimization Pipeline. We optimize and analyze the prompts further using two DSPy optimizers: MIPROv2, SIMBA, and a combination of both. MIPROv2 performs joint optimization of prompt instructions and few-shot examples by generating candidate variations and selecting high-performing combinations using Bayesian optimization. SIMBA introduces stochastic search strategies by sampling multiple execution trajectories and learning from differences between successful and unsuccessful outputs.

[4]Full hyperparameter details are in Appendix A.

3.2 Evaluation Framework.

To represent the semi-flexible nature of IVR systems in our evaluations, we use two filters: a strict rule-based filter, and a flexible LLM judge filter. We then apply conservative scoring to aggregate results robustly.[5].

Rule-based filter. Each response is first evaluated deterministically against ground-truth labels across four match types: action name match (normalized), exact DTMF digit match, speech utterance match (with production-validated synonyms), and alternate response match (historical proven navigation paths). Multiple potential matches are possible within each match type, but a match can only occur for known success cases seen in live calls.

LLM judge. Responses that fail rule-based matching are passed to a two-module LLM judge: one for issue categorization and one for semantic correctness verification. The judge's semantic correctness assertion is accepted only when the assigned category falls within a human-validated allowlist of high-precision categories (e.g., different modality, saying 'agent' vs waiting silently). Categories not in this allowlist are treated as failures regardless of the judge's semantic correctness verdict.

Conservative scoring. Any response not resolved by rule-based matching or a whitelisted judge category is counted as a failure. This conservative design ensures reported accuracy is lower-bounded using production-environment experiments.

3.3 Generalizable Productionization

Cluster-Based Prompt Modularization. A core architectural insight of our pipeline is that IVR nav-

[5]Full category definitions and precision analysis are provided in Appendix B.1.

Dataset	Source Task	Target Task
Train	438	487
Test	-	462
Validation	-	498
Total	438	1,447

Table 1: Prompt Task Transfer Dataset. We collected IVR navigation data in the format of IVR utterance and corresponding AI ground truth actions: 438 pairs for our source task (claims follow up) and 1,447 pairs for our target task (patients' benefits verification). We used all source data as a training set.

Prompt Type	Optimizer	Base (%)	Optim. (%)
Human	MIPROv2	60.82	79.00
Human	SIMBA	60.82	69.48
TL	MIPROv2	45.67	78.14
TL	SIMBA	45.67	81.39
Cluster RAG TL	MIPROv2	**75.92***	**82.21***

Table 2: Prompt optimization results. Cluster RAG Transfer Learning prompts showed statistically significant improvements over best human-designed prompts ($* : p < 0.05$).

igation is semantically heterogeneous: the same model must handle terse DTMF menus, open-ended speech prompts, and multi-step benefit navigation flows present across tasks; each requires distinct response strategies. Encoding all of these in a single monolithic prompt creates a fundamental tension between generality and specificity, which we observed empirically as instruction conflicts and reduced per-cluster accuracy. Our solution, **Cluster-Based Prompt Modularization**, is a deliberate architectural component that decomposes the prompt space along semantic boundaries, enabling targeted optimization per cluster rather than a one-size-fits-all instruction set.

We embed target task IVR utterances using text embeddings and group them into semantically coherent clusters using k-means clustering. For each cluster, the *Transfer Learning Pipeline* is executed independently to produce cluster-specific prompts tailored to the semantic characteristics of the cluster. Then, *Failure Analyzer* identifies systematic failures and missing constraints during evaluation within the *Transfer Learning Pipeline*. We send this analysis to the *Prompt Generation Module*, which makes context aware changes to the prompt. This module makes the prompt more robust in failure scenarios while preserving performance on previously successful cases. It also prevents the number of instructions from drastically increasing while covering those new failure cases. This is important for both cost saving and model performance[6].

Clustered RAG Modeling Pipeline. In our clustered RAG pipeline, each cluster-specific prompt is paired with a retrieval databases for examples and hints tailored to specific IVR utterances. We use text-embedding-based similarity retrieval with a threshold, allowing the system to dynamically add utterance-specific examples and hints during inference. This approach improves robustness and navigation accuracy, as new issues can be accommodated by adding examples to the database rather than modifying prompts or performing extensive re-optimization. Additionally, this design reduces the risk of instruction drift, where prompt changes inadvertently degrade performance.

[6]There are general trends of performance degradations of LLM across different backbone model architectures (Jaroslawicz et al., 2025).

4 Data Description

We collected ground truth IVR navigation data at the turn level, extracting individual (utterance, action) pairs from complete call transcripts across 51 unique outbound payer numbers and 132 distinct call scenarios. The dataset covers the full navigation vocabulary required for healthcare IVR: free-form speech responses, DTMF key presses, structured data inputs (e.g., NPI, patient ID, date of birth, provider tax ID, date of service, callback number, group number), and wait states.

4.1 Stratified sampling

Data splits were constructed using stratified sampling by scenario file rather than global random splits, ensuring all turn types (including rare but critical navigation patterns) are proportionally represented across train, validation, and test sets. Simple random seeding produced imbalanced distributions (e.g., greeting turns disproportionately concentrated in training); stratification corrected this and bolstered the reliability of our evaluation.

4.2 Evaluation Setting

Given a list of acceptable actions, we measure the accuracy of predicting any acceptable action for the given IVR utterance. In real world applications, there can be many acceptable actions for the same IVR utterance. In this section, we discuss and analyze model performance results based on this assumption and cover all possible actions for the

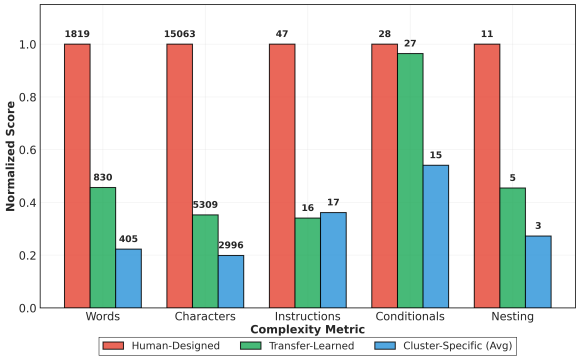

Figure 3: Normalized prompt complexity metrics ($s = x/x_{\max}$) comparing the best human-designed, transfer-learned, and cluster-specific (n=14, averaged) prompts across five dimensions: word count, character count, instructions, conditionals, and nesting depth.

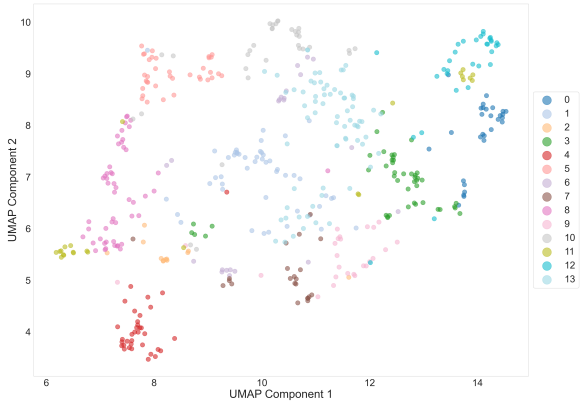

Figure 4: UMAP projection of text embeddings of IVR utterance clusters (see IVR utterance examples for the clusters in Table 11).

given IVR utterances[7].

We first explore different types of prompt optimizer and parameters for this task. Then, we further enhance the pipeline using cluster-based adaptive prompt modeling and show its effectiveness in real-world industrial applications.

We created 'human designed prompts' using heuristics and by incorporating known IVR navigation logic from our source and target tasks. We use these as baselines. Then, we explore both 1) optimization methods and 2) architecture variants.

5 Results

5.1 LLM Prompt Task Transfer

For base prompt performance, human-designed prompts have a higher accuracy than transfer-learned prompts. This is expected, as human-designed prompts include in-context learning (ICL) examples, whereas our transfer learning pipeline removes all ICL examples from the initial task prompt. This design choice prevents human heuristic bias and reduces the risk of hallucinated or irrelevant behaviors being transferred to the target task[8]. After applying DSPy-based optimization, transfer-learned prompts consistently surpass human-designed prompts across optimizers. This suggests that removing ICL examples encourages the optimizer to learn more generalizable instructions which are then specialized for the target task during optimization[9]. Overall, the best performance came from the transfer learning pipeline using the SIMBA prompt optimizer with 81.39% accuracy.

5.2 Cluster Transfer Learning with RAG

In our Cluster RAG model settings, SIMBA degrades performance because stochastic exploration introduces instruction drift and breaks cluster-specific constraints learned through failure analysis. On the other hand, MIPROv2 in light mode performs conservative refinement, preserving previously correct behaviors, making it better suited for stable cluster-level optimization. Cluster RAG model with MIPROv2 obtained the best results for base and optimized prompts and made statistically significant improvements over human-designed model variants.

5.3 Analysis

Modularized Cluster Prompt Analysis. Using our transfer learning pipeline, prompt complexity drastically decreased and accuracy increased compared to a human-designed prompt. Cluster-specific prompts achieved the highest accuracy at 82.21%, outperforming human-designed prompts (79.00%) by 3.2 percentage points despite being 78% shorter (405 vs 1,819 words) and containing 64% fewer instructions (17 vs 47) (Figure 3). The inverse relationship between complexity and performance suggests that simpler, specialized prompts

[7]We provide the exhaustive experiment results considering each evaluation framework component using only one of our evaluation criteria: LLM as a judge, rule matcher and patient information verifier in appendix (see Section A)

[8]To isolate the effect of ICL removal from the effect of automated optimization, we evaluated the human-designed prompt with ICL examples removed. This configuration resulted in 4–5% lower accuracy than the ICL-inclusive baseline, confirming that the manually curated examples encoded genuine domain knowledge rather than heuristic bias.

[9]See more details in Section A

may actually be easier for LLMs to follow accurately, consistent with (Jaroslawicz et al., 2025) showing a negative correlation between instruction count and performance. The two-stage optimization pipeline (transfer learning with clustering) achieved 80% reduction in character count, 73% reduction in nesting depth (from 11 to 3 levels), and 46% reduction in conditionals, while increasing task performance.

Cluster RAG with Dynamic Instructions and Hints. We find that RAG modules help the best IVR navigation model obtain further coverage on non-intuitive edge cases. For example, some payer IVR systems refuse to transfer to an agent until some summary of patient benefits has been provided (Figure 1). The timing of when to request an agent is also not intuitive for LLMs[10]. Among the samples, only our Cluster RAG model made a correct navigation; this model was significantly stronger at capturing similar non-intuitive, timing sensitive responses. Other categories included ambiguous utterances and complex menu options[11].

Generalization across task types. Approximately 75% of utterances in our evaluation set are task-specific to BV IVR. The remaining 25% recur across multiple IVR navigation types (BV, claims, prior authorization), representing universal navigation patterns: end-of-turn decision prompts ("You can ask me another question, or say member / provider"), multi-patient check prompts ("Are there any other patients I can help you with?"), post-benefit-summary navigation loops, and provider credential requests. On these shared utterances, the TL-optimized prompt achieves 80.8% accuracy (consistent with overall benchmark performance) demonstrating that the pipeline captures generalizable navigation patterns beyond the target task.

6 Conclusion

We introduce a generalizable LLM prompt transfer and optimization pipeline that allows scalable task automation in healthcare for IVR navigation. For an industrial production-grade task automation pipeline, it is crucial not only to achieve high performance on the target task benchmark but also to generalize to unseen cases without human effort. Our solution achieves this with a small number of human-verified samples and improves accuracy using an online learning module to retrieve alternate responses without human intervention. We conducted a comprehensive analysis of prompt modularization and measured both offline and online model performances to validate our approaches. We hope future research can leverage our proposed automated LLM task automation pipelines and approaches for automating different AI agent tasks efficiently with minimal human effort.

The results reported in this paper reflect controlled offline and experimental evaluations of specific pipeline configurations; they represent research findings and should not be interpreted as guaranteed production-level performance. The models and prompts evaluated here form one component of a broader production system subject to ongoing monitoring, payer-specific variation, and operational constraints not captured in the experimental setup.

[10] After passing forced benefits navigation loop, some IVR systems don't explicitly provide a menu option for speaking with an agent. For these utterance samples, only Cluster RAG proactively asked for 'Agent' based on the retrieved related cluster prompt, instruction and hints (this response leads to a correct patient benefit department for this payer) while other model variants selected one of explicitly provided menu options from IVR ('next patient' or 'repeat benefits') which can lead to increasing IVR navigation time or even navigation failure.

[11] IVR utterances with more than 50 words on average

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Weize Deng, Yougang Zhao, Jingming Shi, Xiao Chen, Zhe Wang, Ying Qin, Zhiruo Liu, Yuheng Zhang, Dale Schuurmans, and Denny Zhou. 2024. Prewrite: Prompt rewriting with reinforcement learning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128,

Miami, Florida, USA. Association for Computational Linguistics.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: Self-referential self-improvement via prompt evolution. In *International Conference on Machine Learning (ICML)*.

Sherzod Hakimov, Yan Weiser, and David Schlangen. 2024. Evaluating modular dialogue system for form filling using large language models. In *Proceedings of the 1st Workshop on Simulating Conversational Intelligence in Chat (SCI-CHAT 2024)*, pages 36–52, St. Julians, Malta. Association for Computational Linguistics.

Mengze Hong, Wailing Ng, Chen Jason Zhang, Yuanfeng Song, and Di Jiang. 2025. Dial-in LLM: Human-aligned LLM-in-the-loop intent clustering for customer service dialogues. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5885–5900, Suzhou, China. Association for Computational Linguistics.

Megan Hosfield. 2024. Prior authorization and referral process in healthcare and its administration burden. *The Catalyst: Propelling Scholars Forward*, 2:86–106.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Daniel Jaroslawicz, Brendan Whiting, Parth Shah, and Karime Maamari. 2025. How many instructions can llms follow at once? In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling*.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Sergio Rojas-Galeano. 2025. Beyond ivr touch-tones: Customer intent routing using llms. *arXiv preprint arXiv:2510.21715*.

Jiayi Shen, Qi Wang, Zehao Xiao, Nanne Van Noord, and Marcel Worring. 2024. Go4align: Group optimization for multi-task alignment. In *Advances in Neural Information Processing Systems*, volume 37, pages 111382–111405. Curran Associates, Inc.

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. 2023. Multitask prompt tuning enables parameter-efficient transfer learning. In *The Eleventh International Conference on Learning Representations*.

Zifeng Wang, Hanyin Wang, Benjamin Danek, Ying Li, Christina Mack, Luk Arbuckle, Devyani Biswal, Hoifung Poon, Yajuan Wang, Pranav Rajpurkar, et al. 2025. A perspective for adapting generalist ai to specialized medical ai applications and their challenges. *NPJ Digital Medicine*, 8(1):429.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *International Conference on Learning Representations (ICLR)*.

Wenbo Zhao, Arpit Gupta, Tagyoung Chung, and Jing Huang. 2023. SPC: Soft prompt construction for cross domain generalization. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 118–130. Association for Computational Linguistics.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. In *The eleventh international conference on learning representations*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *International Conference on Learning Representations (ICLR)*.

A Hyperparameter Experiments

For all our experiments, we used Gemini 2.5 Flash model for offline training modules, Gemini 2.0 Flash for IVR navigation models and 'text-embedding-004' from Vertex AI for embedding modules. We explored hyperparameters for DSPy (full list available in Table 4) and semantic similarity thresholds based on model performances on the validation set. The best values for hyperparameters are listed on Table 3. We explored two types of DSPy optimizers across our experiments: MIPROv2 and SIMBA. We observed different trends of optimizers depending on the prompt initialization. SIMBA configurations perform better in prompts generated by Transfer Learning (without clusters) because MIPROv2 relies more

Experiment Settings	Best Value
Human prompt DSPy optimizer	MIPROv2
Transferred prompt DSPy optimizer	SIMBA
Number of IVR navigation hints	1
Number of IVR navigation examples	2
Semantic similarity threshold	0.9

Table 3: Selected values for hyperparameters

heavily on early example selection, which can limit exploration when transferred prompts are instruction-centric and data is sparse. In contrast, SIMBA's trajectory-based stochastic search better captures IVR behaviors, leading to superior performance. However, when the initial prompt is human-curated MIPROv2 performs better, as SIMBA's exploration introduces drift in the instruction, as human-curated prompts have been curated with a lot of insights and a precise nature to the task, whereas MIPROv2's example-centric refinement preserves and incrementally improves this structure.

B Automated LLM Improvement Pipeline Components

B.1 Evaluation Framework Modules

LLM as a Judge: As opposed to many LLM benchmarks, in real-world application conversational AI models, there are multiple paths that the models can take for accomplishing the same outcome for their tasks and the action space and possible states are open-ended (e.g., specifically for our task healthcare IVR navigation, we can take different paths to the same department). In order to effectively capture multiple correct answers we used LLM as a judge for action evaluator function. Specifically, this module accept three categories of variations as acceptable responses:

- Speech utterance variations: e.g., both saying 'claim' and saying 'claims' are acceptable for a ground truth response 'claim'.
- Different modality: e.g., for the IVR prompt "For benefits, please say or press 1" and accept both speech utterance saying 'one' or DTMF tone of pressing '1' from phone keypad as a valid correct response.
- Synonymous actions: e.g., IVR utterance "To speak with an agent, say 'agent' or stay on the line", both saying 'agent' or waiting (staying silent) are valid correct responses.

Active Learning Alternate Response Finder: IVR systems can allow different speech utterance variations for each menu option. For example, IVR can say "Welcome to ABC Healthcare. How can I help you today? You can say 'benefit and eligibility', 'claims' or 'prior authorization'." and it accepts both saying 'benefit' and 'benefit and eligibility' to choose 'benefit and eligibility' option. To cover this case, we used 'Alternate Response Finder' that search all actions accepted from IVR navigation historical data in our database[12].

B.2 Prompt Hallucination Checker

'Prompt Hallucination Checker' is a necessary safety component in our system because during the iterative prompt transfer learning, LLM generated prompt can be invalid due to LLM may generating hallucinated action names that are not present in the IVR action space, both of which make the prompt not suitable for production use cases. This issue is further worsened by 'Prompt Similarity Aware Reducer', which aggressively reconstructs the prompt to escape local minima. Even though this module is effective in exploration, occasionally it introduces systemic errors and semantically wrong action names in the prompt. To ensure safety, 'Prompt Hallucination Checker' verifies and validates the prompt, and rejects hallucinated prompts before evaluation or deployment.

B.3 Number of Cluster Analysis

To determine the optimal number of clusters, we evaluated the silhouette score for $k \in \{2, \ldots, 40\}$. Although the score increases as k grows, the rate of improvement gradually diminishes. At k=14, the silhouette score reaches approximately 70% of the maximum value observed at k=40, indicating most of the cluster separations are already captured at this value.

To assess the stability of the structure in each cluster, we analyzed the sample-to-centroid distance distribution at k=14. The distances are concentrated between 0.5 and 0.65, with no heavy tail, which suggests cohesive cluster assignments (Figure 6). Increasing k slightly reduces average distance, however these improvements are mostly achieved by subdividing existing clusters rather than identifying fundamentally new clusters.

[12]We accept an action as valid ground truth 'alternate response' if there is at least one call that reached a target insurance company department.

Prompt Type	Auto	Max Boots. Examples	Max Demos	Max Labelled	Max Steps	Num. Cand.	Optimizer	Temp. Cand.	Temp. Samp.	Val. Acc.	Test Acc.
Human	None	6	-	6	-	36	MIPROv2	-	-	79.72	79.00
Human	Heavy	2	-	6	-	18	MIPROv2	-	-	73.29	72.73
Human	Heavy	4	-	2	-	18	MIPROv2	-	-	77.51	78.57
Human	Medium	4	-	6	-	12	MIPROv2	-	-	66.87	65.37
Human	Medium	6	-	6	-	12	MIPROv2	-	-	69.08	69.05
Human	-	-	6	-	8	9	SIMBA	0.2	0.2	65.86	64.72
Human	-	-	6	-	12	9	SIMBA	0.4	0.4	70.08	68.18
Human	-	-	4	-	12	9	SIMBA	0.4	0.2	69.88	67.75
Human	-	-	6	-	12	6	SIMBA	0.4	0.2	70.88	69.48
Human	-	-	6	-	8	9	SIMBA	0.4	0.4	70.08	68.18
TL	None	6	-	6	-	36	MIPROv2	-	-	75.30	76.19
TL	Heavy	2	-	6	-	18	MIPROv2	-	-	72.49	72.29
TL	Heavy	4	-	2	-	18	MIPROv2	-	-	79.92	78.14
TL	Medium	4	-	6	-	12	MIPROv2	-	-	75.90	76.62
TL	Medium	6	-	6	-	12	MIPROv2	-	-	72.09	72.08
TL	-	-	6	-	8	9	SIMBA	0.4	0.4	74.10	77.49
TL	-	-	6	-	12	6	SIMBA	0.4	0.2	74.50	77.71
TL	-	-	6	-	12	9	SIMBA	0.4	0.4	74.10	77.49
TL	-	-	4	-	12	9	SIMBA	0.4	0.2	74.90	79.87
TL	-	-	6	-	8	9	SIMBA	0.2	0.2	76.91	81.39

Table 4: Hyperparamter Search Results for LLM Prompt Transfer and Optimization Pipeline. The column definitions for DSPy parameters are as follows (parameter names as in the framework): 'Auto': 'auto' parameter for MIPROv2, 'Max Boots. Examples': max_bootstrapped_demos, 'Max Demos': max_demos, 'Max Labelled': max_labelled_examples, 'Max Steps': max_steps, 'Num. Cand.': num_candidates, 'Temp. Cand.' : temperature_for_candidates, 'Temp. Samp.': temperature_for_sampling. 'Acc.' means IVR navigation accuracy (%) for each dataset.

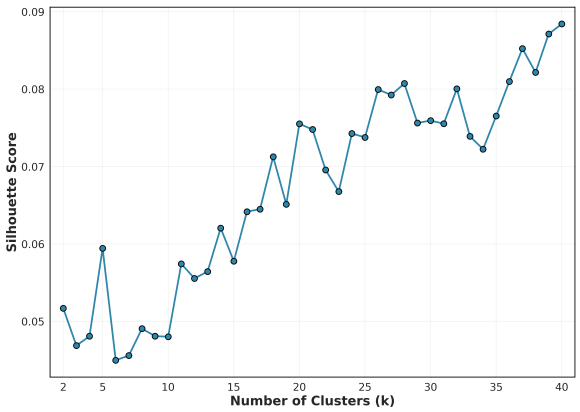

Figure 5: Silhouette scores of $k \in \{2, \ldots, 40\}$ (k = the number of clusters)

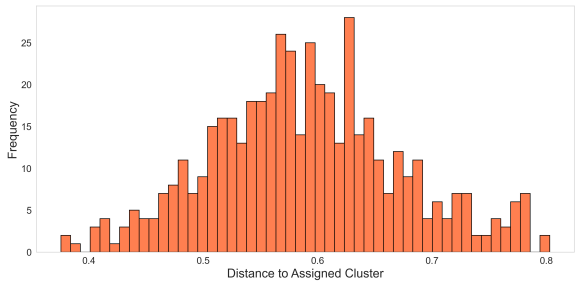

Figure 6: Cluster distance distributions from IVR utterance clusters

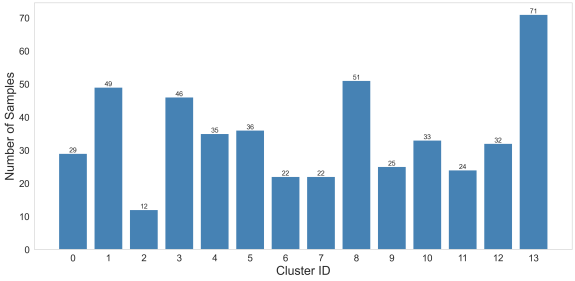

Figure 7: Size of IVR utterance clusters

Cluster size analysis showed that clusters are relatively well balanced, with most containing between 20-50 samples (minimum=12 and maximum=71). This size distribution ensures sufficient data for prompt optimization (see the full results in Figure 7). In contrast, at k=40, many clusters contain fewer than 15 samples, leading to data sparsity. Such small clusters increase the risk of overfitting, as cluster-specific prompts may adapt too closely to examples and fail to generalize.

Therefore, after considering all these factors, we decided to use k=14 as our optimal cluster size and lightweight visual investigation on coherence of cluster samples. You can check the top 5 samples from our final clusters in Table 11

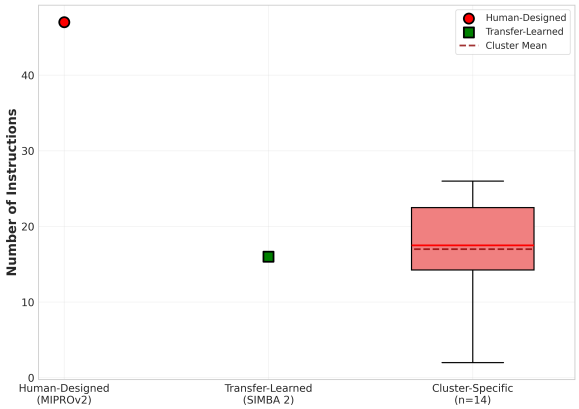

Figure 8: The number of instructions of the best human-designed prompt, transfer-learned prompt and cluster prompts.

B.4 Transfer Learning Pipeline

Transfer learning pipeline has an iterative process with random target task training set sampling and prompt improvement. We provide the detailed workflow and roles of all the components in Figure 9. As an early stopping condition, we tracked the average training accuracy of the five most recent generated target task prompts, compared it with the best five consecutive prompt average, and if it is below the threshold (Epsilon), the tolerance counter is increased or reset. If the tolerance counter is above a certain limit, the iterative loop is terminated. After completing iterations, we evaluate top-K prompts on the validation set and select the highest accuracy prompt.

B.5 IVR Navigation and Hint Database Structure.

IVR navigation example and hint databases consist of 'IVR utterance' as a key and 'additional instruction' and 'hint' as values respectively. At inference time, our pipeline retrieves most similar IVR navigation examples and hints and appends them as a few shot examples and additional instructions to retrieved cluster prompt. For example, for IVR utterance "the benefits quoted were based on the provider's network participation if you would like to receive the contrasting level of benefits say contrasting benefits otherwise say repeat benefit information check another benefit or check pre-authorization requirement by procedure code you can also say next patient", retrieved IVR navigation example was to say 'Customer Advocate' (cosine similarity between the IVR utterance and the key: 0.9572) and hint instruction was 'When the IVR presents multiple, complex navigation options, respond with a general phrase seeking assistance or a human agent.' (cosine similarity between the IVR utterance and the key: 0.9796). In this case, the call can successfully queued for agent when the model responds with 'Customer Advocate' to IVR.

C Prompts of Our Proposed Pipeline Components

Each module in our pipeline uses structured prompting strategies designed to address specific failure modes in iterative prompt transfer. These strategies include meta-prompting with failure-aware adaptation, root-cause analysis, constraint-based validation, similarity-driven compression, controlled exploration, and structured reasoning. All of these strategies helped us to mitigate Transfer bias, Overfitting, Hallucination, Prompt stagnation, Task leakage. To build such prompts, we took inspiration from DSPy's implementation of its modules. The structure of these modules follows the design philosophy of DSPy, in which the prompts are treated as modular components that can be independently refined and optimized. In addition, we build on and customize selected portions of DSPy prompt templates to fit our IVR navigation.

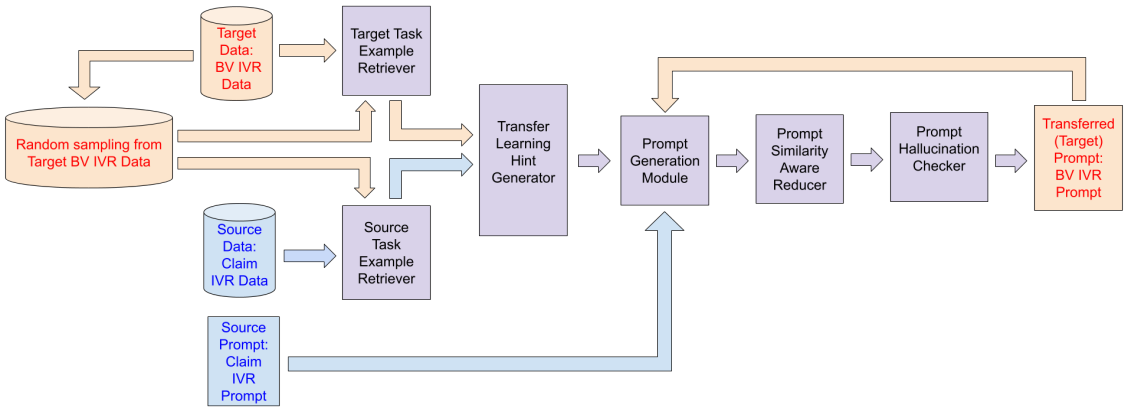

Figure 9: Transfer learning pipeline overall structure. At each iteration, we select random target task IVR navigation utterance samples and 'Target Task Example Retriever' and 'Source Task Example Retriever' collect most similar IVR navigation samples from 'source task training set' and 'target task training set' respectively. Then, they send them to the rest of pipeline. After we get hallucination-free transferred target task prompt, we send it back to 'Prompt Generation Module' and the prompt gets improved with a new set of target task and source task IVR navigation samples.

Prompt	Char. Count	Word Count	Line Count	Inst. Count	Avg. Inst. Length	Max. Inst. Length
Best Human Optimized	15063	1819	77	47	39.72	930
Best TL Optimized	5309	830	34	16	66.94	278
Average of Clusters	2996	405	28	17	35.94	139
Cluster 0	2869	316	47	26	16.58	147
Cluster 1	3162	472	28	16	40.56	97
Cluster 2	2877	364	31	23	17.09	49
Cluster 3	4755	722	26	15	60.93	284
Cluster 4	2451	316	35	24	12.88	64
Cluster 5	2062	255	34	21	17.71	79
Cluster 6	2618	275	41	23	14.70	86
Cluster 7	2142	252	14	8	35.75	90
Cluster 8	4303	658	25	14	59.71	264
Cluster 9	4702	728	23	15	62.33	269
Cluster 10	1707	219	20	12	17.92	49
Cluster 11	2328	258	29	20	15.90	82
Cluster 12	1033	120	11	2	84.00	118
Cluster 13	4936	721	29	19	47.11	262

Table 5: Prompt complexity comparisons for the best optimized human prompt (Best Human Optimized), the best optimized transfer learning prompt (Best TL Optimized), and cluster prompts (Cluster N). 'Char. Count': total number of characters for each prompt, 'Word Count': total number of words for each prompt, 'Line Count': total number of lines for each prompt, 'Inst. Count': total number of instructions in each prompt, 'Avg. Inst. Length': average number of words of all instructions in each prompt, 'Max. Inst. Length': number of the longest instruction from each prompt, 'Average of Clusters': average values each column of all clusters (0-13).

Given the following information, your goal is to **meticulously adapt an existing prompt to create a high-quality, new prompt for Task B for a specific cluster of utterances.** The new prompt should be optimized to guide a Language Model (LM) in accurately navigating IVR systems for this specialized task.

Provided Information:

* **Optimized Prompt for Task A:** A highly effective prompt designed for general IVR navigation (Task A).
* **Task A Description:** A detailed explanation of Task A's objectives and scope.
* **Task B Description:** A clear explanation of the new IVR navigation Task B, highlighting its specific goals.
* **Previous Task B Prompt:** The prompt generated in the prior iteration for this particular cluster of Task B utterances.
* **Task A vs. Task B Hint:** A crucial hint outlining the key differences between general Task A and this specific cluster within Task B.
* **Examples:** Illustrative examples for both Task A and Task B, showcasing expected inputs and outputs.
* **Failed Case Analysis:** A detailed analysis of instances where the **Previous Task B Prompt** failed, including:
* The specific input utterance from Task B that led to the failure.
* The incorrect output generated by the LM using the previous prompt.
* The **reasoning** behind why the previous prompt failed (e.g., misinterpretation of instruction, missing crucial context, ambiguous phrasing, an instruction that led to an incorrect action name).
* The **desired correct output** for that specific failed case.
* Your job is understand the failed cases from previous prompt and generate a new prompt, try to avoid adding examples from failed cases, as it may overfit to the data

Prompt Generation Guidelines:

1. **Adapt and Refine:**
* Leverage the **Optimized Prompt for Task A** as a foundation.
* Incorporate insights from the **Previous Task B Prompt** to understand what has been attempted.
* Utilize the **Task A vs. Task B Hint** to precisely target the differences and guide your modifications.
* Analyze the provided **Examples** for both Task A and Task B to identify patterns, essential information, and critical distinctions that should be reflected in the new prompt.
* **CRITICALLY, learn from the Failed Case Analysis:** Understand *why* the previous prompt failed. Your new prompt should directly address these failure points. If the previous prompt led to misinterpretations or incorrect 'action name' selections, ensure your new instructions prevent similar errors.

2. **Focus and Relevance:**
* **Crucially, remove all instructions, examples, or keywords that are not directly relevant to Task B for this specific cluster of utterances.** The prompt must be laser-focused.
* **Add new instructions as needed** that are specific to Task B for this particular cluster. This is especially important for addressing issues highlighted in the failed case analysis.
* If the overall structure of the prompt can be improved for better performance on Task B, feel free to **restructure it entirely.**

3. **Iteration and Improvement:**
* Your primary objective is to **improve upon the Previous Task B Prompt.** Do not simply repeat it, even if you believe it was previously optimized. Aim for a superior version in each iteration.
* Actively strive to **avoid generating the same prompt repeatedly.** Each iteration should represent a genuine attempt at refinement, particularly in light of failed cases.

4. **Technique Integration (Use one or more):**
* **Creative:** Don't hesitate to think outside the box to craft a more effective instruction.
* **Simple:** Prioritize clarity and conciseness. Avoid unnecessary complexity.
* **Descriptive:** Ensure the instruction is highly informative and provides ample context.
* **High Stakes:** Incorporate a scenario where accurate completion of the task is critical (e.g., "The user's urgent request depends on precise navigation...").
* **Persona:** Assign a relevant persona to the LM (e.g., "You are an expert IVR navigation assistant...").

5. **Action Name Validation:**
* Any 'action name' specified in the generated prompt **must be one of the following**:
{list_of_our_ivr_navigation_action_candidates}.
* **If the prompt contains any other action name, the prompt is considered hallucinated, and you must retry the generation process.**

Table 6: 'Transfer Learning Hint Generator' Module Prompt

Given a prompt, check if the prompt is hallucinated or not. If suppose the prompt contains action name, the action name should be from one of the following: {list_of_our_ivr_navigation_action_candidates}. There can be an option for choosing "Keypad input" for certain actions, like {action_with_keypad_and_speech_option_1}, for all action names except {list_of_free_response_LLM_actions}. If the prompt contains any other action name, then the prompt is hallucinated. If it contains only the action names from the list, then the prompt is not hallucinated.

Table 7: 'Prompt Hallucination Checker' Module Prompt

Given the initial prompt and the rewritten prompt, calculate the similarity score in terms of exact wording between the initial prompt and the rewritten prompt. If the two prompts are highly similar, you have to reduce the length of the rewritten prompt. You can use any technique to reduce the length of the rewritten prompt and return it as reduced length prompt. But you have to make sure the reduced length prompt is still valid and functional. If the two prompts are not highly similar, you don't have to reduce the length of the rewritten prompt. You can return rewritten prompt as reduced length prompt and change structure of the prompt if needed. Be adventurous and creative in your approach to reduce the length of the rewritten prompt. You can take risks by reducing more than what would feel is safe.

Table 8: 'Prompt Similarity Aware Reducer' Module Prompt

Analyzes specific failed instances of an IVR navigation prompt for a particular utterance cluster to identify root causes and suggest areas for prompt improvement. If there are no failed cases, then analysis would be the prompt is working as expected, no changes are needed. Reduce adding more examples from failed cases, as it may overfit to the data. Instead, try to understand the root cause of the failed cases and give your analysis. If there are failed cases due to SSML tags, ignore those failed cases during your analysis, they are issues with the training data, and ground truth not the prompt.

Table 9: 'Failed Case Analyzer' Module Prompt

Analyzes two IVR tasks and their sample IVR utterances to **Objective:** Analyze the provided information for Task A (general) and Task B (specific cluster) to generate a highly detailed and actionable hint. This hint will guide the adaptation of an optimized prompt from Task A to create a specialized prompt for Task B for a particular cluster of utterances.

Input:
* **Task A Description:** Detailed explanation of the general IVR navigation task (Task A).
* **Task A Sample Utterances:** Representative examples of user utterances for Task A.
* **Task B Description:** Detailed explanation of the specific IVR navigation task (Task B) for a particular cluster.
* **Task B Sample Utterances:** Representative examples of user utterances for this particular cluster of Task B.

Hint Generation Guidelines:

1. **Core Purpose:** The hint's primary purpose is to enable the creation of a *cluster-specific* prompt for Task B by leveraging the *optimized general prompt* for Task A.

2. **Detailed Comparison and Contrasts:**
* **Keywords:** Identify and explain the specific keywords, phrases, or entities that are central to Task B (for this cluster) but may be less prominent or absent in Task A, or vice-versa. Highlight any new terminology or unique ways users express their intent in Task B.
* **Intent:** Clearly delineate the precise user intent(s) that Task B aims to fulfill within this cluster, contrasting it with the broader range of intents covered by Task A. Are there narrower, more specific intents in Task B? Or entirely new ones?
* **IVR Path Differences:** Describe how the expected IVR navigation flow or 'action name' sequence for Task B (this cluster) differs from the typical paths in Task A. This includes:
* New or specific 'action name' values required for Task B.
* Differences in the *order* or *combination* of 'action name's.
* Any specific data points (e.g., {task_a_ivr_navigation_action_names}) that are uniquely required or particularly critical for Task B.
* Any 'action name's present in Task A that are *irrelevant* or *should be avoided* for Task B.

3. **Action Name Validation:**
* Any 'action name' referenced within the generated hint **must be one of the following**:
{list_of_our_ivr_navigation_action_candidates}.
* **If the hint contains any other action name, the hint is considered hallucinated, and you must retry the generation process.**

Table 10: 'Prompt Generation Module' Module Prompt

Table 11: Clustering Results of IVR Utterances. Distance: the distance between each utterance and its assigned cluster's centorid.

Cluster	IVR Utterance Sample	Distance
0	office visit the copay is [Dollar Amount] per day	0.5739
0	services are 90% of the allowed amount the calendar year out of pocket for an individual is [Dollar Amount] to change the amount that is [Dollar Amount] the calendar year out of pocket for a family is [Dollar Amount] a year to date the amount met is [Dollar Amount]	0.5790
0	for services provided by Tier 1 provider the member has met individual deductible of [Dollar Amount]	0.5924
0	for this patient's plan the individual deductible is [Dollar Amount]	0.5951
0	office visit the copay is [Dollar Amount] per day The Specialist copay is [Dollar Amount] per day	0.5963
1	please hold while I transfer you to a customer service representative	0.4739
1	thank you for your patience please remain on the line and the next available representative will be with you	0.4828
1	please hold while we transfer you to a customer service professional	0.4965
1	please hold I will transfer you to customer service	0.4989
1	thank you for holding. A representative will be with you as soon as possible if you would like to have the system save your place in line and call you back when an agent is available press 1 otherwise please remain on the line for the next available agent	0.5110
2	all right I see more than one ZIP code for this provider please say the ZIP code where services were rendered	0.3756
2	please tell me the ZIP code where services were rendered	0.4079
2	please say or enter your five or nine digit zip code for your office location	0.4468
2	I'm sorry I did not hear you please say or enter your five or nine digit zip code for your office location	0.4578
2	all right I found more than one address for this provider please select the address from the following list by saying this one when you hear the correct one [Address]	0.4688
3	all right which can I help you with claims eligibility or service benefits you can also say	0.4357
3	which of the following can I help you with you can say verify coverage claim status referral authorization appeal for calling about something else	0.4449
3	thanks I'll just look that up which can I help you with eligibility and benefits claims pre-authorization or other services	0.4509
3	which of the following can I help you with you can say verify coverage claim status referral	0.4593
3	which can I help you with eligibility and benefits claims pre authorization or	0.4604
4	please tell me the patient's date of birth	0.4566
4	what is the patient's date of birth	0.4691
4	date of birth	0.4975
4	sorry what's the patient's date of birth	0.5019
4	what is the date of birth	0.5143
5	the eligibility and benefit lines will be available from 7:30 a.m. to 6 p.m. central time and the claim lines from 8:30 a.m. to 4:30 p.m. central time Out from January 4th 2021 through January 29th 2021 Monday through Friday as a reminder you can quickly obtain patient coverage and claim status online using [Payer Service Website] or your preferred with vendor we are experiencing higher wait times in our service centers as a reminder you are able to obtain routing eligibility and benefit information as well as claim status in seconds using [Payer Service Website] or the web vendor of your choice	0.3204
5	for assistance please hold for the next available representative our normal business hours are Monday through Friday from 8:30 a.m. until 5:00 p.m. Eastern Time	0.3797
5	thank you for calling your call may be monitored or recorded our office is currently closed our regular business hours are Monday through Friday 8 a.m. to 6 p.m. Eastern standard Time please call back during these hours or you can find answers to many of your questions online by using our automated web tools at [Payer Service Website] thank you for calling and have a good day	0.3952

Continued on next page...

Table 11 – continued from previous page

Cluster	IVR Utterance Sample	Distance
5	through January 24th 2020 Monday through Friday claims lines hours remain the same 8:30 a.m. until 4:30 p.m. central time as a reminder you are able to obtain routing eligibility and benefit information as well as claim status in seconds using [Payer Service Website] or the web vendor of your choice in order to get eligibility or benefits will need your rendering NPI or HMO site number for claims or any other inquiries will need your billing NPI patient now what is your 10-digit NPI or HMO site number	0.3976
5	please listen carefully as our menu options have changed this call may be recorded for quality purposes [Payer Name] is currently experiencing high call volume for your convenience [Payer Name] provides self-service options online via our portal at [Payer Service Website] or by our automated touch-tone system available 24 hours a day 7 days a week if you still require assistance please stay on the line and your call will be answered in the order it was received	0.4074
6	This call may be monitored or recorded for quality and training [Payer Name] offers self-service functionality and required provider to use [Payer Portal] or phone automation for routine inquiries please say or enter your part number	0.4945
6	this call may be monitored or recorded for Quality For assistance with your or a family member's Health Plan say I'm a member otherwise in a few words tell me why you're calling today	0.5218
6	thanks for calling [Payer Name] this call may be monitored and recorded for quality assurance to ensure that you are getting the information you need in a quick and convenient way we provide digital solutions available 24/7 go to [Payer Service Website] for self-service options such as member eligibility and benefits claim status and claim appeals for our interactive voice response system say or enter your tax identification number	0.5312
6	thank you for calling [Payer Name]'s dedicated provider service center to improve our service your call will be monitored and recorded by continuing with this call you understand accept and agree that the information communicated by [Payer Name] is not an offer of payment does not guarantee coverage or payment and a subject to all Benefit Plan terms and conditions including member eligibility at the time of service please say or enter your NPI or tax ID	0.5364
6	thanks for calling [Payer Name] provider services this call may be monitored or recorded for quality assurance purposes for self-service options including member eligibility and benefits check please go to [Payer Service Website] to begin please say or enter your 10-digit NPI number	0.5516
7	you have entered [Phone Number] if this is correct press one if this is incorrect press 2	0.3621
7	you said [Phone Number] if that's right say yes or press 1 otherwise say no or press 2	0.3800
7	I'm sorry let's try again using your telephone keypad please enter the ten digit phone number starting with the area code	0.4206
7	let's try that again please enter or say the subscriber's ID excluding any letters	0.4357
7	please say or enter the subscriber's ID including any letters	0.4580
8	for which patient can I also have the patient's first and last name	0.3848
8	okay and what's the patient's name	0.4371
8	please say the patient's first and last name	0.4705
8	first let's find the member please say the member's ID including all letters	0.4800
8	please tell me the patient's first and last name	0.4908
9	this quote has been saved to a file and your confirmation number is [Number] now you can say repeat that or benefit details you can also say next patient or main menu, if you're through go ahead and hang up	0.3639
9	this quote has been saved to a file and your confirmation number is [Number] now you can say repeat that or	0.3705
9	pre authorization is not required for [Product Code] and [Product Code] plan a reservation required minutes have been saved to a file your confirmation number is [Number] would you like me to fax these pre authorization requirements to you	0.3803
9	pre authorization is required through [Payer Name] for [Product Code] and finally pre authorization is not required for service [Code] these pre authorization requirements have been saved to a file your confirmation number is [Number] would you like me to fax these pre authorization required to you	0.3957
9	pre authorization is not required for [Product Code] and finally pre authorization is required through [Payer Name] for [Address] please pre authorization have been saved to a file your confirmation number is [Number] would you like me to fax these pre authorization requirements to you	0.3995

Continued on next page...

Table 11 – continued from previous page

Cluster	IVR Utterance Sample	Distance
10	hello thank you for calling the [Payer Name] provider line my name is [Name] how can I help you today	0.4496
10	thank you for calling [Payer Name] this is [Name] how may I help you	0.4578
10	thank you for calling [Payer Name] my name is [Name] how can I help you	0.4632
10	thank you for calling [Payer Name] military and Veterans Health Care Claims Administrator for access to all of your needs visit us on our website at [Payer Service Website]	0.4842
10	hi thank you for calling the [Payer Name] provider line this is [Name] can I please have your first and last name	0.4851
11	eligibility and benefits first what's the member ID	0.4836
11	if it's excluding the three character prefix what's the subscribers ID	0.4862
11	eligibility and benefits first what's the subscriber ID	0.4891
11	okay first what's the member's ID	0.5284
11	first what's the member's ID	0.5404
12	please be advised that a quote of eligibility and benefits is not a guarantee of payment all benefit payments are subject to eligibility medical necessity and the terms conditions limitations exclusions and payment levels of the patient health benefit plan at the time the services are rendered benefit payments are usually not determined based on patient ID charges and might be significantly less than Bill charges please note newborn dependents not listed on the membership file may have benefits available this patient covered under a PPO plus plan coverage began [Date] to pre-existing the 3-character prefix is [Prefix] and the group number is [Group Number]	0.4063
12	please be advised that a quote of eligibility and benefits is not a guarantee of payment all benefit payments are subject to eligibility medical necessity and the terms conditions limitations exclusions and payment levels of the patient's health benefit plan at the time the services are rendered	0.4400
12	please be advised that a quote of eligibility and benefits is not a guarantee of payment all benefit payments are subject to eligibility medical necessity and the terms and conditions limitations exclusions and payment levels of the patient health benefit plan at the time the services are rendered benefit payments are usually not determination based on billed charges and might be significantly less than billed charges please note newborn dependents not listed on the membership file may have benefits available	0.4513
12	all right eligibility and benefits verification of benefits or coverage is not a guarantee of eligibility or payment which is subject to continued eligibility and timely payment of premium wage actual payment is based on the terms and conditions of the plan all claims are subject to review upon submission all benefits quoted reflect in network level benefits unless otherwise requested are you calling for eligibility status or some	0.4942
12	[Payer Name] verification of a member's benefits is not a guarantee of payment and [Payer Name] is not entering into a contract for payment of any amount by providing this information payments are owned and when claims are received and processed through the members plan do you need information for eligibility PCP deductible or coinsurance you should also say give me a summary	0.5282
13	your call may be recorded for quality assurance press 1 to verify eligibility or obtain a member's ID number press two to obtain benefit information press three for claims inquiries press 4 for in patient [Provider Name] and Post Acute admissions medical prior authorization and clinical appeals press 5 for Tom inquiries press 6 all others press seven to repeat these options press pound	0.4626
13	if you are a security health plan member please press one now to be transferred to a customer service representative if you are calling for durable medical equipment press two if you're calling in regards to any prior authorization press three if you are calling for prescription benefits press four if you are calling for eligibility benefits claim status name or processing questions press five	0.4784
13	to check claim status eligibility and benefits or for network related information press one for pre certification and authorization press 2 for behavioral health and substance use press 3 to repeat these options press 9	0.4860
13	please listen carefully before choosing your selection for questions regarding dental press 2 for prescription pre-authorizations press three for claims or eligibility information wage not including pre authorization press four for questions around mental health chemical dependency our hemophilia medication press five for questions around [Payer Name] plan a hospice press 6 all other calls remain on the line	0.4862

Continued on next page...

Table 11 – continued from previous page

Cluster	IVR Utterance Sample	Distance
13	sorry I didn't hear you to hear that again say repeat that or press 1 for benefits say benefit details or press two for eligibility for someone else plan a next patient or press three for anything else say main menu or press four	0.4935