To Chat or Task: a Multi-turn Dialogue Generation Framework for Task-Oriented Dialogue Systems

Daniel Rim[†,‡] **Minsoo Cho**[†] **Changwoo Chun**[†] **Jaegul Choo**[‡]
[†]Hyundai Motor Company [‡]KAIST AI
{drim,minsoocho,cwchun}@hyundai.com
jchoo@kaist.ac.kr

Abstract

Task-oriented dialogue systems employ natural language understanding (NLU) modules to manage the intricate and continually evolving business requirements of production systems. Although the development of Large Language Models (LLMs) introduced extraordinary chitchat capabilities, implementing LLMs into such systems brought new difficulties. One of the main challenges is the lack of specific datasets for training and evaluation of systems that offer both capabilities: chat and task. As NLU modules are designed to handle complex task requests and LLMs are utilized to specifically answer chitchat interactions, the system must correctly identify the functional intent of the user to utilize an applicable module. This paper presents CTFusion, a multi-turn dialogue generation framework designed to assist the evaluation and training of production systems that offer both capabilities. Utilizing the framework, we generate a multi-turn dialogue dataset for an in-vehicle speech recognition system, which includes 41,211 dialogues of 240 real-world in-vehicle intents, and train In-vehicle Context Sensor (ICS), a lightweight model that successfully identifies the functional intent of the driver. ICS outperforms all baseline models across various experimental settings, which demonstrates that CTFusion can help generate relevant datasets with a complex business logic, which can subsequently assist production systems in leveraging LLMs for their chitchat capabilities.

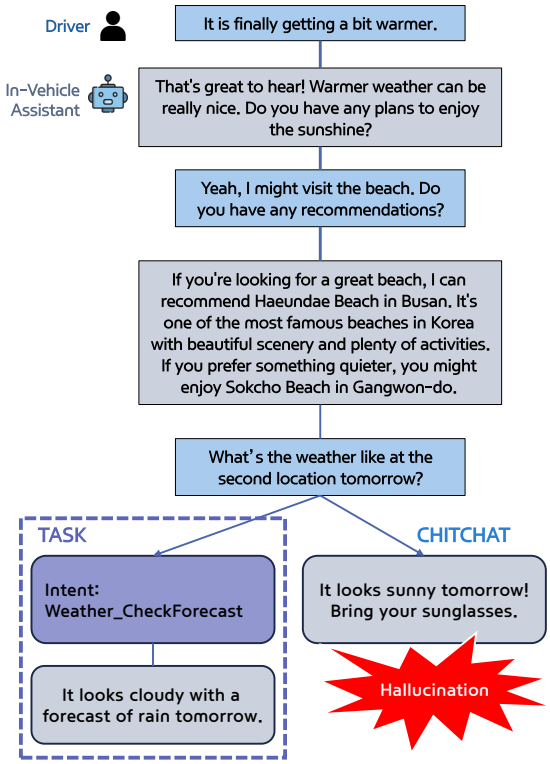

Figure 1: Motivation for functional intent classification. Checking the weather forecast is one of many tasks that the NLU module of in-vehicle assistant is designed to handle, as it utilizes real-time information from external tools to answer the driver's request. If the last utterance is incorrectly recognized as a continuation of "chat", a LLM-powered agent is likely to hallucinate, as it is only designed for chats. If identified as "task", the NLU module utilizes relevant tools to respond properly.

1 Introduction

Recently, the incorporation of large language models (LLMs) in various smart assistants enhanced the overall user experience, enabling users to engage in complex dialogues with the assistant. Pretrained on a substantial amount of text, LLMs provide improved natural language understanding, enable better context awareness, and reduce dependency on rigid predefined scripts for more dynamic and intuitive interactions (Radford et al., 2019; Zhang et al., 2019; Brown et al., 2020). The release of ChatGPT (OpenAI, 2022), and other open-source models, such as Llama (Meta, 2024), Phi-4 (Abdin et al., 2024), along with various orchestration modules, such as LangChain (Topsakal and Akinci, 2023) and AutoGen (Wu et al., 2023), made the integration of such models simpler.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 576–592

The latest in-vehicle speech recognition (IVSR) systems also utilize LLMs (Rony et al., 2023; Mathis et al., 2024) to handle chitchat, allowing drivers to have natural conversations with the in-vehicle assistant. Implementing LLMs in production environments, however, presents considerable challenges. Traditionally, task-oriented systems were built to understand a single utterance from a user, without any conversational capabilities. They employed a natural language understanding (NLU) module, which is connected to external tools and APIs, to manage the intricate and continually evolving business requirements of production systems. For example, IVSR systems utilize a NLU module to understand and respond to a driver's task requests, such as open the window, set the temperature, or turn on the radio. As demonstrated in Figure 1, if a driver asks a question that requires the assistant to check the weather, the system must identify the functional intent as task, and utilize the NLU module to answer the request. If the system fails to recognize the task, a likely response is a hallucination, as the relevant information is not available to the LLM agent.

Although LLMs are capable of much more than traditional NLU modules, it is widely accepted that LLMs cannot completely replace the existing modules. (Yi et al., 2024). More specifically, with 240 specific intents that must be recognized as a task intent in IVSR systems, no available LLMs are able to guarantee production-level requirements in accuracy and latency. For any production-level task-oriented system to offer LLM-powered chitchat capabilities without performance decline, it must be able to identify the functional intent of utterances, and leverage both modules for their respective purposes. Given the specificity of this scenario, it is unsurprising that no datasets specifically designed for this purpose are available.

In this work, we introduce CTFUSION, a dataset generation framework, which generates dialogues that can facilitate the training and evaluation of task-oriented systems that offer chitchat capabilities. Our goal is to provide a pipeline that can be adapted to any specific needs of task oriented-systems, as production assistants are not all alike and offer a different set of tasks and chitchat capabilities. CTFUSION first utilizes system-specific tasks to generate intent-slot sets and action sequences, which provide the foundation for dialogue generation. To further ground our work, the framework uses seed utterances from real user dialogues. After generating based on the foundation, the dialogues go through further augmentation to introduce more diversity in the dataset.

Utilizing our pipeline, we generate IVSR-CTF, which has 41,211 Korean dialogues with an average of 8.5 turns for 240 real-world in-vehicle driver intents. We limit the dialogue pattern to always transition from chitchat to task, as the dialogue ends once a task is identified and completed by IVSR systems. Based on this dataset, we train In-vehicle Context Sensor (ICS) to demonstrate the applicability of CTFUSION. ICS demonstrates production ready results in all experimental settings for functional intent classification, addressing the need to identify the functional intent of each utterances.

Overall, the major contributions of our work are as follows:

- We introduce CTFUSION, a dataset generation framework for multi-turn dialogues with chitchat and task requests between an assistant and a user. It is designed to generate realistic dialogues with minimal human effort, to help train and evaluate systems that employ both capabilities.
- We empirically demonstrate the applicability of CTFUSION in IVSR systems by generating IVSR-CTF, an in-vehicle specific dialogue dataset, and training ICS, a lightweight model for functional intent classification.

2 Related Work

2.1 Existing IVSR Systems

Prior to the development of LLMs, IVSR systems typically handled single-turn commands by processing user inputs through intent classifiers and slot extractors (Lim et al., 2022). These systems are capable of handling simple tasks, but are not designed to handle multi-turn dialogues, where the intent can be omitted from the last utterance from the driver. (Ferreira Cruz et al., 2020) For example, when a user asks, "What's the weather in Seoul today?" followed by, "How about tomorrow?" the system fails to capture key contexts like "Seoul" or "weather" without explicit mechanisms for handling multi-turn dialogues (Hindle and Rooth, 1993).

After the release of LLMs, some proposed methods in implementing such models in IVSR systems. BMW proposed CarExpert, an in-car conversational question answering module based on

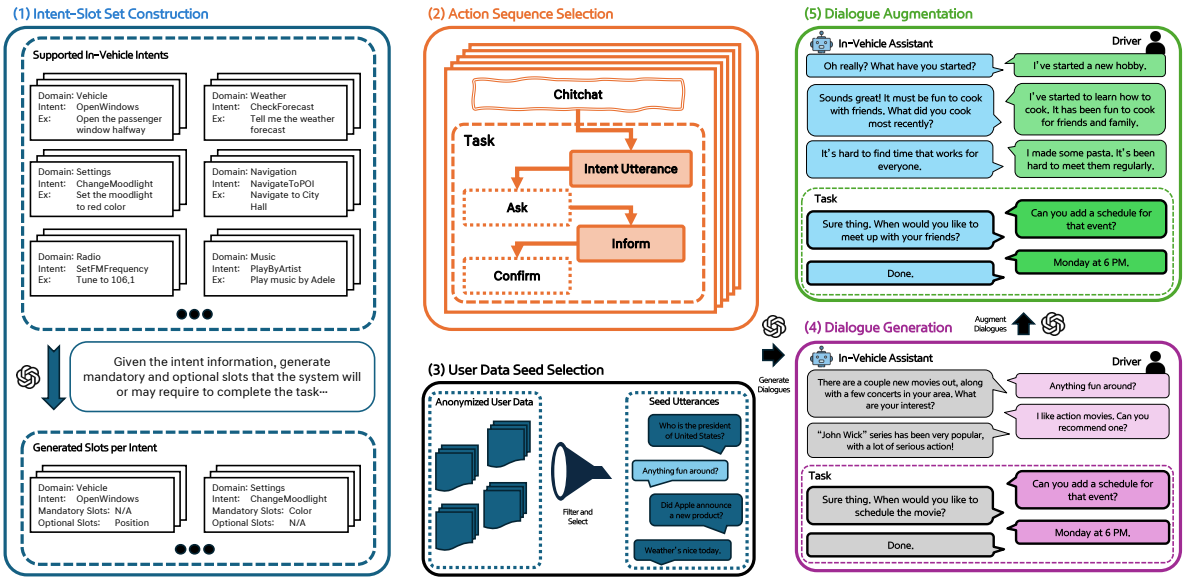

Figure 2: Overview of CTFUSION, our multi-turn dialogue generation pipeline: 1) **Intent-Slot Set Construction**: a list potential mandatory and optional slots are generated with GPT-4o, 2) **Action Sequence Selection**: potential action sequences are selected for the given intent; 3) **User Data Seed Selection**: real user utterances are randomly selected as seeds for dialogue generation; 4) **Dialogue Generation**: dialogues are generated based on the previous steps; and 5) **Dialogue Augmentation**: dialogues are further augmented for diversity.

retrieval-augmented generation (RAG) (Rony et al., 2023). Although RAG-based agents can be beneficial in reducing hallucinations, CarExpert does not handle the functional intent changes in conversations, staying in "chat" mode during the session that requires LLM-based answers. Others have developed hybrid architectures that takes the advantage of the strengths of LLMs, while limiting their downsides (Chun et al., 2025). Our research aligns with the utilization of a hybrid architecture; however, rather than employing a GPT-4o model to identify the functional intents in driver utterances, we develop a framework for generating a dataset, and train a lightweight model for the same purpose. This approach effectively reduces overall production costs by avoiding additional LLM requests.

2.2 Chitchat-Task Integration in Dialogue Systems

Research on management of task-oriented dialogues with chitchat have relied on the MultiWOZ dataset (Budzianowski et al., 2018; Zang et al., 2020), and its variants, such as FusedChat (Young et al., 2022) and InterfereChat (Stricker and Paroubek, 2024). Although the incorporation of task-oriented and chitchat dialogues together aligns with our research, these datasets only include a significantly smaller number of intents, dealing with at most 11 intents. IVSR systems require a much more fine-grain intent classification, where there are 240 intents that must be accurately identified. Furthermore, these datasets are not in-vehicle specific, where the conversation follows a very distinct distribution. Our work introduces a pipeline that can be adapted to generate a dataset for any task-oriented systems. Details of comparable dialogue datasets are shown on Table 1.

Others suggest generating datasets for task-oriented dialogues, which are only applicable for systems that process user queries as tasks. Some utilize schema-guided process for the generation of dialogues (Shim et al., 2025; Lee et al., 2022; Kale and Rastogi, 2020; Rastogi et al., 2020), where dialogue sequence is predefined prior to the generation. We elect to utilize a similar mechanism in outlining the overall dialogue prior to generation, but also include chitchat interactions to expand the potential application of the framework.

Some researchers propose a proactive unified model designed to capture the potential need for a switch from chitchat to task-oriented services with a transition info extractor (Liu et al., 2023b). The model then utilizes a transition sentence generator to seamlessly recommend task services to the user. While such an approach can be suitable for some task-oriented dialogue systems, it is not directly applicable to IVSR systems, which prioritize fulfill-

Datasets	SalesBot 2.0	FusedChat	IVSR-CTF
Seed Data	SalesBot 1.0	MultiWOZ	Real Driver Data
Domain	General	General	In-Vehicle Specific
No. Intents	6	11	240
No. Dialogues	5,453	10,436	41,216
Average Turns	7.71	18.36	8.57

Table 1: Dialogue dataset statistics. IVSR-CTF is specifically generated for the IVSR domain.

ing user requests rather than suggesting new tasks. Moreover, extending interactions by introducing extra turns in dialogues is discouraged in IVSR systems, as erroneous recommendations can lead to a worse user experience. Although our framework can be modified to incorporate proactive interactions between the user and the system, we focus specifically on in-vehicle scenarios to demonstrate its applicability to IVSR systems.

3 CTFusion

In this section, we present CTFUSION, our dataset generation framework. The overview can be seen in Figure 2, and the data generation process is described in detail in the subsequent sections. The details of the IVSR-CTF, an IVSR system specific dataset generated with our pipeline, can be found in Table 1 and Figure 3. We include example prompt templates in Appendix E.

3.1 Generation Pipeline

Intent-Slot Set Construction To generate a natural dialogue that includes functional intent changes from chitchat to task-oriented dialogues, we first generate a list of mandatory and optional slots for each task intent. This enables the generation process to incorporate slot filling conversations into the dialogue. We prompt GPT-4o (Hurst et al., 2024) to generate relevant slots for the given intent, and classify them as mandatory and optional.

Action Sequence Selection We observe that to generate dialogues that follow the distinct interaction pattern of a target system, it is necessary to predefine the sequence of utterances. For a given intent, we construct dialogue action sequences by setting the length of the chitchat, and the flow of task utterances. For instance, to design the task utterance interactions, check the dialogue intent type. If the action sequence is predefined to have a "complete" dialogue intent type, the intent utterance is prompted to include all mandatory slot values. If it is "incomplete", the intent utterance lacks some mandatory slots, and the task utterances include slot filling utterances between the assistant and the user. Lastly, the assistant "confirms" the task request to conclude the dialogue. The action sequence outlines the dialogue, allowing the framework to have a finer control over the generated dialogues. We outline various action sequences, and select one for generation based on the number of mandatory slot values for the task intent.

User Data Seed Selection We notice that the generated data from GPT-4o can be very monotonous. To promote diversity and factuality, the seed utterance that starts the dialogue is randomly selected from real user utterances. For example, in the case of IVSR systems, since in-vehicle conversations follow a very distinct style, the seed driver utterances guide the generation process to output authentic interaction patterns.

Dialogue Generation We prompt GPT-4o with a simple instruction to generate a realistic dialogue based on the intent-slot set, action sequence, example utterance of the intent, and the seed utterance. GPT-4o generates the assistant utterance based on the given seed utterance, then continues to generate based on the action sequence, ending with a confirmation from the assistant to conclude the dialogue.

Dialogue Augmentation Although seed utterances promote some diversity, we identified that dialogue topics were too limited. Therefore, we systematically augment the generated dialogues to promote diversity in the dataset. For each intent, we first identify various topics in the chitchat dialogues based on Latent Dirichlet allocation (Blei et al., 2003). Once the topics are identified, we prompt a LLM to generate different potential topics that could be relevant to the intent. We then prompt GPT-4o to alter the dialogue by switching the topic of the dialogue, while maintaining the user's intent in the task utterances. Lastly, we alter the length of the dialogues by modifying the number of chitchat and task utterances, while maintaining the overall contents of the dialogue.

3.2 Dataset Details

With CTFUSION, we are able to generate IVSR-CTF, a diverse dialogue dataset that is based on real user utterances. We repeat the process to generate over 150 appropriate dialogues per each intent.

Dataset Quality To assess the quality of IVSR-CTF, we sampled 80 dialogues across all domains, and evaluated them using G-Eval (Liu et al., 2023a)

and 5 human annotators, who are knowledgeable of IVSR systems. Inspired by the evaluation metrics from Shim et al. 2025, the following criteria on a 3 point scale were used to evaluate:

- *Naturalness*: Is the chitchat dialogue **natural** between a driver and IVSR assistant?
- *Coherence*: Are the generated utterances from the driver and the assistant **coherent** with the dialogue context?
- *Efficiency*: Are the assistant's utterances in the dialogue **efficient**?

	G-Eval	Human Eval
Naturalness	2.56	2.45
Coherence	2.76	2.80
Efficiency	2.93	2.85

Table 2: Evaluation results of IVSR-CTF.

Table 2 shows the average scores from G-Eval and human annotators. Both G-eval and human annotators assigned high scores to the dialogues across all three criteria. This indicates that the generated dialogues from CTFUSION are natural, contextually coherent, and efficiently designed. We also include G-eval scores for all dialogues for each domain in Appendix C.

4 Methodology

We define the problem setting to validate CTFUSION and its applicability in a production setting.

4.1 Problem Definition

Given a dialogue sequence from IVSR-CTF, the goal of functional intent identifier is to correctly classify the intent of driver utterance. Similar to that of SimpleTOD (Hosseini-Asl et al., 2020), which was originally designed for task-specific scenario, we redefine the objective by adapting it for a functional intent classification; chat or task. We explicitly label the dataset to chat or task mode, which represent the functional intent of each utterance. The dialogue data is incrementally fed to the model, including the previous dialogue history, and the goal is to classify the current driver utterance.

4.2 In-vehicle Context Sensor

We train **I**n-vehicle **C**ontext **S**ensor (ICS) by instruction fine-tuning a Llama-3.2-3B-Instruct (Dubey et al., 2024) to identify the functional intents of utterances in in-vehicle dialogues. We select this model as the base model, as the goal is to utilize the smallest model possible for a solution that can improve the IVSR system. Without additional fine-tuning, models smaller than Llama-3.2-3B-Instruct, such as Llama-3.2-1B-Instruct or Kanana Nano 2.1B (Bak et al., 2025), showed significant drop in following instructions in identifying the functional intents. In Algorithm 1, the overall IVSR system procedure is outlined. ICS classifies the functional context of the current utterance. If it is classified as "chitchat", the LLM-powered chitchat module responds, generating a natural response. If it is classified as "task", the NLU module processes the utterance and performs the requested task. Accurately classifying the functional intent is crucial, as each module is dedicated to each functional intent.

Algorithm 1 IVSR System Procedure

Input: H: Dialogue History, U: Driver Utterance, LM: LLM Module, ML: ML Module
Output: A: System Answer, T: System Task Action
Function IVSR(H, U)**:**
  $U_{text} \leftarrow ML_{ASR}(U)$ // speech to text
  $D \leftarrow LM_{ICS}(H, U_{text})$ // determine context
  if D *is chat* **then**
    $R \leftarrow LLM_{chat}(H, U)$ // generate response
    $T \leftarrow null$ // no task for chat
  end
  else
    $R \leftarrow ML_{NLU}(H, U)$ // generate response
    if R *has a task associated* **then**
      $T \leftarrow ML_{task}(R)$ // perform task T
    end
  end
  $A \leftarrow ML_{TTS}(R)$ // transform R to answer A
  return A, T

5 Experiments and Results

In these experiments, we evaluate ICS in identifying functional intents of utterances in multi-turn dialogues between a driver and an in-vehicle assistant. The input to the model is a dialogue history, which can be represented as the following:

$$H_n = (u_1, s_1, u_2, s_2, ..., u_n, s_n) \quad (1)$$

where H_n is the dialogue history up to the n-th turn, and u_i and s_i are the utterances from the driver and the assistant. We split IVSR-CTF into training, validation, and test sets in roughly an 8:1:1 ratio. Specifically, we use about 30k dialogues for training, 4k dialogues for validation, and 4k dialogues for testing. We also leave out 24 intents from the

Models	Test Set		Unseen Intents		Real Driver Data	
	Acc.	F1 Score	Acc.	F1 Score	Acc.	F1 Score
Phi-4-14B	64.71%	0.769	67.13%	0.796	66.13%	0.742
EXAONE 3.5-32B	70.05%	0.811	69.97%	0.815	65.85%	0.752
GPT-4o Mini	79.06%	0.875	81.38%	0.894	78.96%	<u>0.850</u>
GPT-4o	<u>82.62%</u>	<u>0.899</u>	<u>84.63%</u>	<u>0.915</u>	<u>79.51%</u>	0.839
Llama-3.2-3B	53.68%	0.674	48.36%	0.632	62.30%	0.730
ICS (OURS)	**90.36%**	**0.908**	**90.72%**	**0.919**	**82.51%**	**0.874**

Table 3: Performance of various LLMs on the identifying the functional intent of driver utterances. The classification accuracy and F1 score is reported. The best results are in **bold**, while the second best are <u>underlined</u>.

training, corresponding to approximately 4k dialogues, for an additional evaluation.

5.1 Evaluation Tasks and Metrics

Given the dialogue history, shown on Equation 1, the task is to classify the current u_i from the driver. Each driver utterance is labeled based on the history up to that turn, but no labels are included in the dialogue history. The model is prompted to identify the functional intent of the current utterance. We measure accuracy and F1 score of functional intent classification, which can either be "chat" or "task".

Along with the test set, we also include two more evaluations: Unseen Intents and Real Driver Data. As production systems are updated with new features, new intents are constantly introduced. We leave out 24 intents as unseen intents from the dataset to evaluate the model's adaptability, simulating a likely scenario where new intents are introduced. Furthermore, we evaluate our model on 366 real driver utterances in 93 dialogues. These utterances are manually labeled by two external human annotators, and were not used as seed utterances during the generation process.

5.2 Baselines

We compare ICS with the following baseline models. We select GPT-4o (Hurst et al., 2024) and GPT-4o-mini (OpenAI, 2024), which represent the best available chat models. As our dataset is in Korean, we also select EXAONE 3.5-32B (An et al., 2024) and Phi-4 (Abdin et al., 2024) models, to represent multi-lingual LLMs. Lastly, we compare ICS with Llama-3.2-3B-Instruct model to investigate the impact of the training process.

5.3 Experimental Results

Test Set Results Looking at the results on Table 3, it is clear that the GPT-series has the upper hand on non-finetuned models. As for the multi-lingual language models, EXAONE 3.5 demonstrated suitable results, outperforming Phi-4 models. ICS demonstrates the best results, outperforming all other models. This supports the notion that in a complex scenario, without fine-tuning, base LLMs with in-context reasoning cannot guarantee production-level requirements (Yi et al., 2024). When comparing ICS with the Llama-3.2-3B-Instruct model, it is clear that the finetuning process on IVSR-CTF significantly improved the functional intent classification performance.

Unseen Intents & Real Driver Data Results For any solution to be production-ready, it must be robust to updates to the system. To simulate such situations where new intents are introduced, we measure the performance of all models for the 24 unseen intents. All models show equivalent performance, even showing a slight improvement in performance. Although the intents were not included in the training process, ICS demonstrates robust performance in such simulated setting. ICS also exhibits the best results on the real driver data, indicating that the CTFUSION properly generates realistic dialogues for the target domain. Full results for each domains can be found in the Appendix D.

5.4 Ablation Study: Augmentation

We evaluate the impact of the augmentation in CTFUSION by training a separate model on the generated dataset that were not processed with augmentation. As shown in Table 4, though ICS without augmentation performed relatively well, outperforming all other baseline models on synthetic data, it showed a significant drop on the real driver data. Without the augmentation step, we speculate that the patterns of the generated dialogues are not diverse enough to capture the subtleties that define in-vehicle conversations. This further proves that to build a model that can generalize to real-world

scenarios, the factuality and fidelity of the synthetic data must be ensured (Liu et al.). We believe that refining the augmentation process could be an area of research that could further improve the dataset generation pipeline.

ICS	w/ Augmentation		w/o Augmentation	
	Acc.	F1 Score	Acc.	F1 Score
Test Set	90.36%	0.908	85.07%	0.914
Unseen Intents	90.72%	0.919	87.83%	0.915
Real Driver Data	82.51%	0.874	62.30%	0.570

Table 4: Augmentation Analysis of ICS.

6 Conclusion

In this work, we introduce CTFUSION, a pipeline for generating a multi-turn dialogue dataset for integration of LLMs with task-oriented systems. With the proposed pipeline, we generate IVSR-CTF, a multi-turn dialogue dataset, and train ICS to identify functional intents of the driver within a multi-turn dialogue. ICS demonstrates the applicability of CTFUSION, which allows us to accurately assess the functional intent of the driver. Furthermore, CTFUSION can be modified for other task-oriented assistants with chitchat capabilities, assisting in the training and evaluation process of such systems. Although IVSR-CTF is limited to a chitchat to task pattern, different action sequences can be designed for other systems. For example, one could design action sequences for a smart home assistant that include more transitions, such as chitchat to task to chitchat, or task to another task to chitchat, etc. These findings are particularly relevant for systems that are starting to incorporate LLMs, as the pipeline generates appropriate synthetic datasets, facilitating the addition of chitchat capabilities without any degradation to the core task performance.

Limitations

Although CTFUSION generates applicable datasets for task-oriented systems, several limitations remain that highlight areas for future improvement.

LLM Selection The LLM used to generate plays a critical role in overall quality of the dataset. Our goal was to generate a Korean dataset, and therefore we elected to use GPT-4o in various parts of the framework. When attempted with a smaller model, the generated dataset did not meet the quality requirements. Applying CTFUSION to other languages might require other models, as there may be more appropriate models for different languages. Evaluating other LLMs for different languages, and further optimizing the generation process remains an important future direction.

Limited Augmentation Methods Although augmentation improved the quality of the dataset, we were unable to perform multiple types of augmentation for additional analysis. Choice of topic modeling methods could have a significant impact on the augmentation process. As this showed promising results, we leave this as future work, potentially comparing various methods in generating high factuality and fidelity data.

Dependency on Well-defined Specifications As CTFUSION utilizes predefined intents, their descriptions, and example utterances during generation, it heavily relies on the quality of system specifications. This could limit the potential use, as not all system specifications are well-defined.

Dataset Due to the nature of in-vehicle conversations, the action sequences always followed a sequence of chat to task, without additional transitions. Depending on the nature of dialogues and system requirements, the action sequences can be refined for the specific needs. As IVSR-CTF and experiments on ICS are performed on real user data, we are unable to provide more details regarding the dataset. Unfortunately, we are not able to release IVSR-CTF to the public, as it contains specific details regarding the IVSR system design. However, CTFUSION can be utilized for other domains to generate domain specific datasets.

Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.RS-2019-II190075 Artificial Intelligence Graduate School Program (KAIST)), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-00555621), and Hyundai Motor Company.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Soyoung An, Kyunghoon Bae, Eunbi Choi, Kibong Choi, Stanley Jungkyu Choi, Seokhee Hong, Junwon Hwang, Hyojin Jeon, Gerrard Jeongwon Jo, Hyunjik Jo, et al. 2024. Exaone 3.5: Series of large language models for real-world use cases. *arXiv e-prints*, pages arXiv–2412.

Yunju Bak, Hojin Lee, Minho Ryu, Jiyeon Ham, Seungjae Jung, Daniel Wontae Nam, Taegyeong Eo, Donghun Lee, Doohae Jung, Boseop Kim, et al. 2025. Kanana: Compute-efficient bilingual language models. *arXiv preprint arXiv:2502.18934*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

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. *Advances in neural information processing systems*, 33:1877–1901.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Changwoo Chun, Daniel Rim, and Juhee Park. 2025. LLM ContextBridge: A hybrid approach for intent and dialogue understanding in IVSR. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 794–806, Abu Dhabi, UAE. Association for Computational Linguistics.

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

André Ferreira Cruz, Gil Rocha, and Henrique Lopes Cardoso. 2020. Coreference resolution: toward end-to-end and cross-lingual systems. *Information*, 11(2):74.

Donald Hindle and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational linguistics*, 19(1):103–120.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191.

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

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6505–6520.

Harrison Lee, Raghav Gupta, Abhinav Rastogi, Yuan Cao, Bin Zhang, and Yonghui Wu. 2022. Sgd-x: A benchmark for robust generalization in schema-guided dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10938–10946.

Jungwoo Lim, Suhyune Son, Songeun Lee, Changwoo Chun, Sungsoo Park, Yuna Hur, and Heuiseok Lim. 2022. Intent classification and slot filling model for in-vehicle services in korean. *Applied Sciences*, 12(23):12438.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. Best practices and lessons learned on synthetic data. In *First Conference on Language Modeling*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Ye Liu, Stefan Ultes, Wolfgang Minker, and Wolfgang Maier. 2023b. System-initiated transitions from chit-chat to task-oriented dialogues with transition info extractor and transition sentence generator. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 279–292, Prague, Czechia. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Lesley-Ann Mathis, Can Günes, Kathleen Entz, David Lerch, Frederik Diederichs, and Harald Widlroither. 2024. Generating proactive suggestions based on the context: User evaluation of large language model outputs for in-vehicle voice assistants. In *Proceedings of the 6th ACM Conference on Conversational User Interfaces*, pages 1–7.

AI Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. *Meta AI Blog. Retrieved December*, 20:2024.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8689–8696.

Md Rashad Al Hasan Rony, Christian Suess, Sinchana Ramakanth Bhat, Viju Sudhi, Julia Schneider, Maximilian Vogel, Roman Teucher, Ken Friedl, and Soumya Sahoo. 2023. CarExpert: Leveraging large language models for in-car conversational question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 586–604, Singapore. Association for Computational Linguistics.

Jeonghoon Shim, Gyuhyeon Seo, Cheongsu Lim, and Yohan Jo. 2025. Tooldial: Multi-turn dialogue generation method for tool-augmented language models. *arXiv preprint arXiv:2503.00564*.

Armand Stricker and Patrick Paroubek. 2024. A few-shot approach to task-oriented dialogue enhanced with chitchat. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 590–602.

Oguzhan Topsakal and Tahir Cetin Akinci. 2023. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. In *International Conference on Applied Engineering and Natural Sciences*, volume 1, pages 1050–1056.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024. A survey on recent advances in llm-based multi-turn dialogue systems. *arXiv preprint arXiv:2402.18013*.

Tom Young, Frank Xing, Vlad Pandelea, Jinjie Ni, and Erik Cambria. 2022. Fusing task-oriented and open-domain dialogues in conversational agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11622–11629.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

A Implementation Details

We use Llama-3.2-3B-Instruct (Dubey et al., 2024) with LoRA (Hu et al., 2022) to efficiently fine-tune the model while reducing memory overhead. Instead of full fine-tuning, we apply LoRA adaptation with rank 16, LoRA scaling factor $\alpha = 16$, and a dropout rate of 0.01. We optimize the model using Paged AdamW (Loshchilov and Hutter, 2019) with a learning rate of 2e-4, a weight decay of 0.001, and gradient clipping at 0.3. The training is conducted with a batch size of 4 per GPU and gradient accumulation of 1 step. We train for 5 epochs, scheduling a warm-up ratio of 3%, and use constant learning rate decay. All experiments are conducted with four NVIDIA A6000 GPUs.

B Domain Names

Domain names and distribution can be found in Table 5 and Figure 3.

Domain Label	Names	Intents	No. Dialogues
A	Vehicle Control	91	15860
B	Map and Navigation	28	4814
C	General Information and Queries	26	4461
D	Media Control	21	3491
E	Built-In Camera Control	16	2783
F	Weather Information	13	2242
G	Volume Control	12	2057
H	Bluetooth Control	9	1534
I	Cluster Information	7	1208
J	Payment and Transactions	4	687
K	Schedule Management	4	671
L	USB Control	3	494
M	Help	3	465
N	Phone Control	3	449
Total		240	41216

Table 5: Domain names and the number of intents and dialogues in the IVSR-CTF.

C G-Eval Results for All Dialogues

G-eval results for all dialogues in IVSR-CTF can be found in Table 6.

D Full Domain Results

Full domain results for test set and the unseen intents can be found in Table 7 and Table 8.

E Prompts

We display the prompt templates used to generate slots for each intent in Figure 4, and dialogues in Figure 5, as well as the prompt template used to augment the generated dialogues in Figure 6. We also include the prompt template used to identify the intent of the driver's utterance in Figure 7.

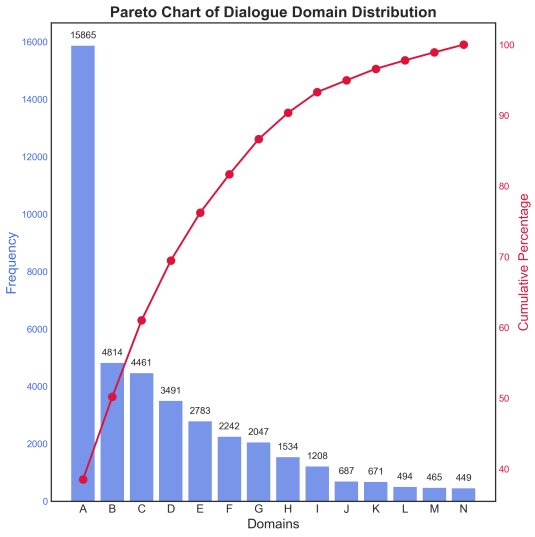

Figure 3: Domain distribution of the dialogues in IVSR-CTF.

F Dataset Examples

We show two example dialogues from IVSR-CTF in Figure 8 and Figure 9. As the dialogues are all in Korean, they were translated into English for demonstration.

G-Eval Results															
	Average	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
Naturalness	2.52	2.43	2.56	2.34	2.61	2.48	2.51	2.66	2.29	2.37	2.53	2.45	2.68	2.25	2.59
Coherence	2.59	2.53	2.66	2.44	2.71	2.58	2.61	2.76	2.39	2.47	2.63	2.55	2.78	2.45	2.69
Efficiency	2.90	2.83	2.91	2.88	2.95	2.86	2.92	2.97	2.84	2.89	2.93	2.85	2.98	2.87	2.94

Table 6: G-eval results for all dialogues in IVSR-CTF.

Test Set Accuracy															
Models	**Total**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
Phi-4	64.71%	65.42%	55.24%	72.96%	69.46%	56.08%	85.98%	55.34%	45.95%	67.24%	64.29%	62.07%	51.85%	80.77%	76.00%
EXAONE 3.5-32B	70.05%	63.06%	72.58%	87.12%	77.25%	64.86%	90.65%	55.34%	60.81%	68.97%	67.86%	65.52%	70.37%	84.62%	88.00%
GPT-4o Mini	79.06%	77.24%	73.39%	84.55%	83.23%	80.41%	91.59%	72.82%	71.62%	77.59%	78.57%	75.86%	88.89%	88.46%	88.00%
GPT-4o	82.62%	83.71%	77.42%	85.41%	86.83%	81.76%	89.72%	69.90%	71.62%	77.59%	92.86%	86.21%	85.19%	92.31%	88.00%
Llama-3.2-3B	53.68%	43.78%	55.65%	71.67%	67.07%	39.19%	81.31%	55.34%	43.24%	43.10%	50.00%	58.62%	55.56%	76.92%	84.00%
ICS (OURS)	90.36%	90.82%	89.87%	88.84%	93.21%	86.21%	93.20%	93.94%	83.56%	92.98%	92.86%	89.29%	92.31%	91.67%	83.33%
Test Set F1 Score															
Models	**Total**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
Phi-4	0.769	0.781	0.688	0.829	0.806	0.697	0.920	0.690	0.621	0.802	0.766	0.762	0.675	0.880	0.863
EXAONE 3.5-32B	0.810	0.765	0.833	0.924	0.856	0.778	0.949	0.696	0.742	0.806	0.793	0.775	0.815	0.910	0.934
GPT-4o-mini	0.875	0.865	0.837	0.909	0.900	0.885	0.954	0.833	0.824	0.861	0.877	0.857	0.941	0.936	0.929
GPT-4o	0.899	0.907	0.866	0.917	0.926	0.896	0.942	0.810	0.829	0.868	0.962	0.921	0.919	0.959	0.934
Llama-3.2-3B	0.674	0.592	0.702	0.816	0.792	0.546	0.888	0.695	0.566	0.580	0.665	0.712	0.711	0.860	0.908
ICS (OURS)	0.908	0.907	0.914	0.901	0.928	0.873	0.913	0.954	0.886	0.892	0.952	0.898	0.953	0.925	0.838

Table 7: Full domain mode classification accuracy for test set.

Unseen Intents Accuracy												
Models	**Total**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**
Phi-4	67.13%	69.24%	61.11%	68.50%	66.39%	66.67%	92.24%	57.23%	47.13%	52.35%	45.35%	75.76%
EXAONE 3.5-32B	69.97%	64.39%	70.76%	86.71%	76.35%	79.89%	93.10%	52.60%	49.43%	46.47%	54.65%	88.48%
GPT-4o Mini	81.38%	80.98%	77.48%	89.30%	83.61%	86.20%	96.55%	65.89%	75.86%	65.88%	60.46%	92.72%
GPT-4o	84.63%	86.22%	77.49%	87.57%	83.61%	89.08%	95.98%	65.32%	81.61%	84.12%	73.84%	87.88%
Llama-3.2-3B	48.36%	40.46%	48.83%	69.36%	59.54%	52.30%	77.59%	43.35%	25.29%	14.12%	22.09%	72.12%
ICS (OURS)	90.72%	90.29%	93.31%	89.70%	93.00%	86.34%	89.12%	93.33%	90.59%	91.62%	93.29%	86.62%
Unseen Intents F1 Score												
Models	**Total**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**
Phi-4	0.796	0.816	0.753	0.810	0.791	0.800	0.960	0.728	0.641	0.687	0.624	0.862
EXAONE 3.5-32B	0.815	0.782	0.825	0.929	0.861	0.888	0.964	0.689	0.662	0.635	0.707	0.939
GPT-4o Mini	0.894	0.894	0.870	0.943	0.908	0.926	0.982	0.794	0.863	0.794	0.754	0.962
GPT-4o	0.915	0.925	0.872	0.934	0.908	0.942	0.979	0.790	0.899	0.914	0.849	0.935
Llama-3.2-3B	0.632	0.571	0.648	0.819	0.744	0.687	0.874	0.605	0.404	0.247	0.362	0.838
ICS (OURS)	0.919	0.921	0.925	0.851	0.974	0.883	0.918	0.924	0.892	0.904	0.980	0.893

Table 8: Full domain mode classification accuracy for unseen intents. The 24 left out intents only included 11 domains, compared to 14 total in IVSR-CTF.

[System]
You are tasked with generating relevant slots for the given intent and description.

For some driver intents, they need slot values for the system to complete the task. For example, for the intent of adding schedules, the system must know the specific date and time, which is a required slot. There may be optional slots, such as the name of the meeting, meeting type, or who are attending the meeting. Another example would be where the intent is setting the temperature of the fatc of the vehicle. In this case, the temperature and the specific zone can both be optional, as the vehicle is capable of just turning on the fatc function.

You are to generate some mandatory slots and optional slots for the intent.
You will be given some example slots, of which can both be optional or mandatory.
You do not have to include the example slots.

Output the slots in the following format:
Mandatory = []
Optional = []

[User]
The intent of the driver for this conversation is {intent}.
Here is a description about the intent: {description}
Here is an example task utterance: {task}.
Here are some potential slots for the intent: {slot}

[Assistant]

Figure 4: Prompt template for generating slots for each intent.

[System]
You are to generate a Korean dialogue between an in-vehicle speech recognition assistant and a driver.
The driver's utterance should be marked either "Chit Chat" or "Task" for the mode of the utterance to determine the intent of the driver.
Driver's utterances should be kept short and informal, without using excessive instructions.

You will be given a "seed" utterance, which should start the conversation.
This portion of the conversation should be marked as "Chit Chat".
Strictly maintain the driver's request to a chitchat type interaction to emulate a lighthearted conversation between a driver and the system.

You will be given an "intent" of the dialogue. The goal of the dialogue is for the driver to utter a task-oriented message with the given intent that requests the system of a task associated with the intent. The task-oriented message can refer to the previous utterances with coreferences.

You will be given an "example" utterance of the task. Only utilize the example utterance as guidance and generate a different task utterance with the same intent for the dialogue.

You will be given an "action sequence" of the dialogue. The dialogue should follow this action sequence, in which the sequence of utterance types are defined.
In the action sequence, the "intent task utterance" tag is where the driver requests a task-oriented message with the given intent. This utterance should be marked with the intent.

You will be given a list of "slots" for the intent. The mandatory slots should be included in the task portion of the dialogue.
If the action sequence for the dialogue has "complete" for the intent type, the intent task utterance should include all information about the mandatory slots.
If the action sequence for the dialogue has "incomplete" for the intent type, the dialogue should follow the task utterance with assistant asking for slot information, and the driver giving the slot information. These should be marked as "ask" and "inform" intents.
Feel free to include information about the optional slots in generating the task utterances from the driver.

Make sure the conversation transition is consistent with the dialogue topic and natural.
Here is an example with navigation to a specific poi:

```
"dialogue": [
  {
    "role": "driver",
    "mode": "Chit Chat",
    "text": "What is there to do in Busan?",
    "intent": null
  },
  {
    "role": "assistant",
    "mode": "Chit Chat",
    "text": "I recommend visiting the Bosu-dong Book Street in Busan. You can purchase a variety of used books at affordable prices, and there is also a cultural festival held every October.",
    "intent": null
  },
  {
    "role": "driver",
    "mode": "Chit Chat",
    "text": "Do they only sell used books there?",
    "intent": null
  },
  {
    "role": "assistant",
    "mode": "Chit Chat",
    "text": "At Bosu-dong Book Street, they sell not only used books but also various new releases. Additionally, there are many cafes and restaurants nearby, so you can enjoy a relaxing time reading books.",
    "intent": null
  },
  {
    "role": "driver",
    "mode": "Chit Chat",
    "text": "What's the most popular restaurant there?",
    "intent": null
  },
  {
    "role": "assistant",
    "mode": "Chit Chat",
    "text": "The most popular restaurant at Bosu-dong Book Street is Ijaemo Pizza's main branch.",
    "intent": null
  },
  {
    "role": "driver",
    "mode": "Task",
    "text": "Alright, let's go there.",
    "intent": "Navigation_NavigateToPOI"
  },
  {
    "role": "assistant",
    "mode": "Task",
    "text": "Okay, I'll guide you there.",
    "intent": "Confirm"
  }
]
```

Once the driver expresses the intent above, and all the mandatory slots of the intent are filled, the dialogue ends with the assistant's response, with the "Confirm" intent.
Make sure the format of the dialogue follows the example.

[User]
The intent of the driver is {intent}.
Here is a seed utterance: {seed}.
Here is an example task utterance: {task}.
The mandatory slots of the intent is {mandatory}.
The optional slots of the intent is {optional}.
Here is the action sequence: {action}.

[Assistant]

Figure 5: Prompt template for generating dialogues for each intent. The example is translated into English for demonstration.

[System]
For the following dialogue between a driver and an AI assistant in a car, you are to alter the dialogue to improve diversity of dialogues. Do not alter the personality or their specific roles when applying this update.

The driver is always talking informally towards the assistant, without really including all valid information. The assistant is a helpful assistant in a vehicle, looking to answer questions and performing specific tasks as requested by the driver.

You are to first identify the topic of the chitchat turns in the dialogue and update the chitchat portion to the given new topic. Design the dialogue to naturally transition towards the task portion of the dialogue.

Additionally, you are to do one of the following:
1. Reduce the number of chitchat turns, without making the dialogue unnatural.
2. Increase the number of chitchat turns in the beginning.
Make sure to update the existing chit chat turns to ensure smooth transition.

Output the updated dialogue in the same format as the input.

[User]
Dialogue: {dialogue}
New Topic: {topic}

[Assistant]

Figure 6: Prompt template for augmenting generated dialogues.

[System]
For the following dialogue, you are to determine if the intent of the last utterance from the driver is task oriented or chit chat.

You will be given a list of task-oriented intents, example utterances, and their descriptions.

If the last driver utterance is task oriented based on the dialogue, and is one of the intents, output "Task".

Task oriented can mean one of the two following things:
1. The assistant is requested to perform an action in the car, such as controlling the infotainment system or other features in the car.
2. The assistant is requested to find external information, such as current weather forecast, sports event scores, or perform a function that requires connection to external tools.

For example,
Driver: I don't know why the weather is so hot these days.
Assistant: It's really hot these days. You can't live without air conditioning.
Driver: Exactly, without air conditioning it'd be a real problem.
Since the last utterance is NOT task oriented, and is of a Chit Chat type of utterance, the output would be "Chit Chat".

Driver: I don't know why the weather is so hot these days.
Assistant: It's really hot these days. You can't live without air conditioning.
Driver: Exactly, without air conditioning it'd be a real problem.
Assistant: Totally, especially in the car—it's even worse.
Driver: True, at least you can roll down the windows to cool off in the car.
Assistant: Exactly. But it's nice to keep the windows open for a breeze—it feels pretty refreshing.
Driver: Should we close the windows now?
Since the last utterance is task oriented with one of the intents from the list, the output would be "Task".

Just output "Task" or "Chit Chat". No reasons or any explanations.

[User]
Intents: {intent_descriptions}
Dialogue: {dialogue}

[Assistant]

Figure 7: Prompt template for identifying driver utterance's mode. The example is translated into English for demonstration.

```
"dialogue": [
        {
          "role": "driver",
          "mode": "Chit Chat",
          "text": "Is there anything fun happening these days?",
          "intent": null
        },
        {
          "role": "assistant",
          "mode": "Chit Chat",
          "text": "Recently, several movies and dramas have been trending. What genre do you like?",
          "intent": null
        },
        {
          "role": "driver",
          "mode": "Chit Chat",
          "text": "I like action movies. Do you have any recommendations?",
          "intent": null
        },
        {
          "role": "assistant",
          "mode": "Chit Chat",
          "text": "In that case, the 'John Wick' series, which was recently released, has been popular. It has plenty of amazing action
scenes!",
          "intent": null
        },
        {
          "role": "driver",
          "mode": "Task",
          "text": "Will it be cold tomorrow?",
          "intent": "Weather_CheckIfCold"
        },
        {
          "role": "assistant",
          "mode": "Task",
          "text": "After checking tomorrow's weather, it looks like it will be chilly in the morning and a bit warmer in the afternoon,
but it might still feel cold. It's better to wear slightly thicker clothes when heading out.",
          "intent": "Confirm"
        }
      ]
```


Figure 8: Dataset example for the *Weather_CheckIfCold* intent. The example is translated into English for demonstration.

```
"dialogue": [
        {
    enjoyable "role": "driver",
            "mode": "Chit Chat",
            "text": "Recently, I've started a new hobby.",
            "intent": null
        },
        {
            "role": "assistant",
            "mode": "Chit Chat",
            "text": "Oh, really? What hobby did you start?",
            "intent": null
        },
        {
            "role": "driver",
            "mode": "Chit Chat",
            "text": "I'm learning to cook. Cooking with my friends is so fun.",
            "intent": null
        },
        {
            "role": "assistant",
            "mode": "Chit Chat",
            "text": "That sounds great! Cooking with friends must be really enjoyable. What dishes have you tried making?",
            "intent": null
        },
        {
            "role": "driver",
            "mode": "Chit Chat",
            "text": "I tried making pasta. But it's so hard to schedule time with everyone.",
            "intent": null
        },
        {
            "role": "assistant",
            "mode": "Chit Chat",
            "text": "That's true, coordinating schedules can be tough. Still, plans with friends are important, so you should definitely
make them happen.",
            "intent": null
        },
        {
            "role": "driver",
            "mode": "Task",
            "text": "Actually, I've arranged to meet a friend this weekend. Please add that to my schedule.",
            "intent": "AddSchedule_AddEvent"
        },
        {
            "role": "assistant",
            "mode": "Task",
            "text": "This weekend? What time are you meeting? Please provide the 'date and time.'",
            "intent": "Ask"
        },
        {
            "role": "driver",
            "mode": "Task",
            "text": "Saturday afternoon at 3:00 PM.",
            "intent": "Inform"
        },
        {
            "role": "assistant",
            "mode": "Task",
            "text": "Got it, I will add the appointment to your schedule.",
            "intent": "Confirm"
        }
    ]
```


Figure 9: Dataset example for the *AddSchedule_AddEvent* intent. The example is translated into English for demonstration.