Towards Effective Long Conversation Generation with Dynamic Topic Tracking and Recommendation

Trevor Ashby[1], Adithya Kulkarni [1], Jingyuan Qi[1], Minqian Liu[1], Eunah Cho[2], Vaibhav Kumar[2], Lifu Huang[1]
[1]Virginia Tech, [2]Amazon
{trevorashby, aditkulk, jingyq1, minqianliu, lifuh}@vt.edu
{eunahch,kvabh}@amazon.com

Abstract

During conversations, the human flow of thoughts may result in topic shifts and evolution. In open-domain dialogue systems, it is crucial to track the topics discussed and recommend relevant topics to be included in responses to have effective conversations. Furthermore, topic evolution is needed to prevent stagnation as conversation length increases. Existing open-domain dialogue systems do not pay sufficient attention to topic evolution and shifting, resulting in performance degradation due to ineffective responses as conversation length increases. To address the shortcomings of existing approaches, we propose EVOLVCONV. EVOLVCONV conducts real-time conversation topic and user preference tracking and utilizes the tracking information to evolve and shift topics depending on conversation status. We conduct extensive experiments to validate the topic evolving and shifting capabilities of EVOLVCONV as conversation length increases. Un-referenced evaluation metric UniEval compare EVOLVCONV with the baselines. Experimental results show that EVOLVCONV maintains a smooth conversation flow without abruptly shifting topics; the probability of topic shifting ranges between 5%-8% throughout the conversation. EVOLVCONV recommends 4.77% more novel topics than the baselines, and the topic evolution follows balanced topic groupings. Furthermore, we conduct user surveys to test the practical viability of EVOLVCONV. User survey results reveal that responses generated by EVOLVCONV are preferred 47.8% of the time compared to the baselines and comes second to real human responses.

1 Introduction

The responses in open-domain dialogue systems are designed to align with the intentions of human users (Chen et al., 2017). Due to the human flow of thoughts, human intentions and requirements evolve as the conversation progresses (Klinger, 2014). Therefore, topic evolving and shifting is necessary for dialogue systems to maintain a long and engaging conversation with users.

Recently, Ma et al. (2024) proposed a clustering system with a self-training autoencoder to detect in-domain topics in an unsupervised manner, and Wu et al. (2024) proposed uncertainty and diversity-based sampling techniques to identify topics of interest from extracted topics efficiently. These recent works focus on identifying and reusing the topics discussed in the conversation with no scope for evolving the topics. Sevegnani et al. (2021) adapted text generation models to generate responses that bridge the new topic to the topic of the previous conversation turn. This approach has scope for topic evolvement; however, it is very restrictive since it can only handle one-turn topic transitions and requires the next response as input. None of these approaches explicitly model user preferences.

In this study, we overcome the shortcomings of previous studies by proposing EVOLVCONV, which conducts dynamic topic tracking and user preference analysis for better topic evolving and shifting. Specifically, EVOLVCONV includes a topic-tracking module that captures implicit and explicit conversational cues, extracts discussed topics from the conversation, and user preferences for the topics. A Graph structure is used to efficiently store the extracted topics, their relationships, and user preferences that serve as a conversation history tracker. EVOLVCONV takes advantage of the graph structure to retrieve potential topics and user preferences that can be part of generated responses. However, these retrieved topics are already discussed in the conversation; therefore, EVOLVCONV includes a topic recommender module that recommends novel topics aligning with the retrieved topics for better topic evolving and shifting. Finally, a response generation module generates responses with the recommended topics

Proceedings of the 17th International Natural Language Generation Conference, pages 540–556
September 23–27, 2024.

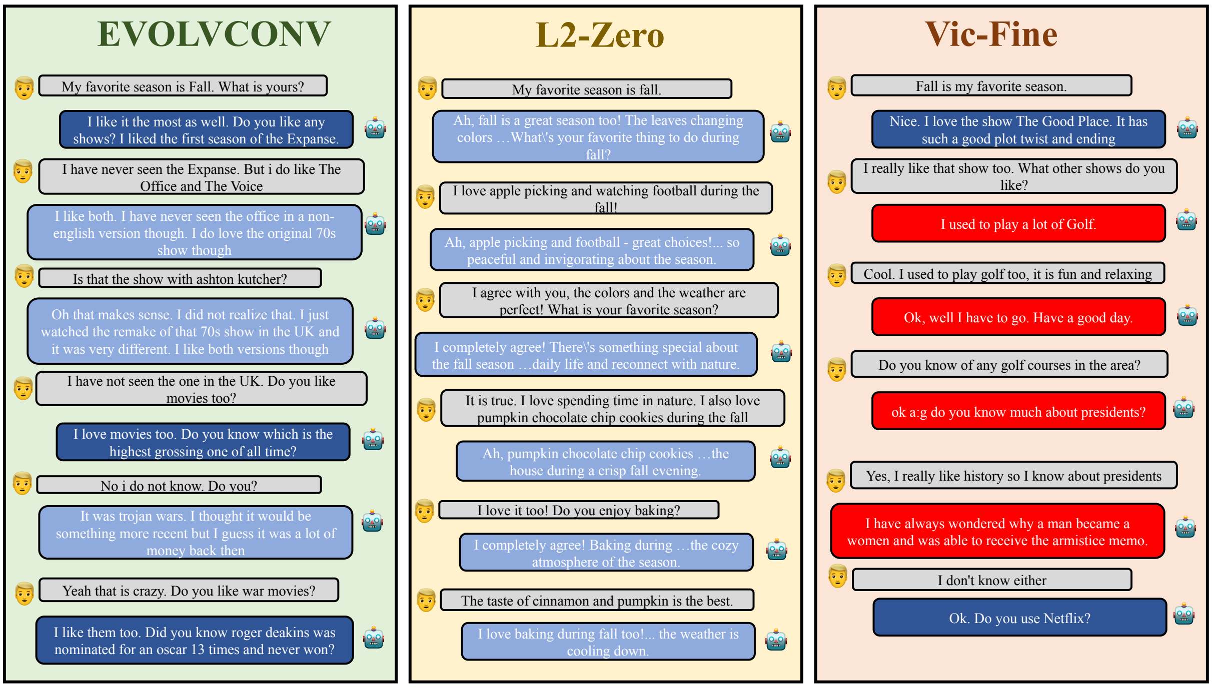


Figure 1: Comparing conversation experience with EvolvConv compared to the baselines. **Bold blue** and **Bold red** represent in-context and out-of-context topic shifting/evolution triggered by the model. We can observe from the conversations that EvolvConv can perform topic shifting and evolving more effectively than the baselines.

that align with user preferences. Figure 1 compares the conversation experience of EVOLVCONV with the baselines. The conversation demonstrates that EVOLVCONV can evolve topics and smoothly shift between related topics, keeping the conversation interesting and the user engaged.

Due to our proposed novel architecture, existing datasets cannot be used for training EVOLVCONV. Therefore, we propose new datasets to train modules of EVOLVCONV. We train EVOLVCONV on our proposed new datasets and evaluate on benchmark datasets to test the topic-shifting and evolving capabilities of EVOLVCONV. Specifically, we evaluate (1) the topic-shifting probability, (2) how well the topic evolves in responses generated, and (3) user preference modeling for long conversations. The experimental results show that EVOLVCONV balances topic-shifting and evolving better than the baselines. Specifically, EVOLVCONV does not hastily shift topics during initial turns in the conversation, and topic shifting is done based on an understanding of user requirements as the conversation progresses. Similarly, EVOLVCONV provides sufficient discussion time for each topic and then smoothly evolves to new topics. We conduct user surveys to analyze the user preference modeling capabilities of EVOLVCONV. The survey reveals that the responses generated by EVOLVCONV are preferred by the users for long conversations. Overall, the experimental results confirm that the dynamic topic tracking and recommendation capabilities of EVOLVCONV result in effective long conversation generation.

To summarize, the following are the key contributions of this work.

(1) We propose a conversation history tracker that extracts topics and user preferences for the topics from conversation utterances and stores them as a graph structure.

(2) Our proposed topic recommender focuses on better topic evolving and shifting by recommending topics that align with the current conversation turn.

(3) Our proposed response generator takes advantage of the advancements in LLMs to generate responses preferred by users.

(4) We propose topic tracking and topic recommendation datasets for model training.

2 Related Work

Conversation Topic and User Preference Tracking: Understanding the topics of the conversation and user preferences for the topics can help generate effective and relevant responses. Unsupervised studies in conversation topic extraction in Open Domain Dialogue (ODD) propose augmenting temporal relationship information between responses with TF-IDF-based vector space model (Adams and Martell, 2008) or applying Latent Dirichlet allocation (LDA) model for topic extraction (Yu and Xiang, 2023). Earlier supervised approaches trained logistic classifiers, support vector machines, and gated recurrent units (Park et al., 2019) to extract topics. Recently, Zhang et al. (2020) proposed a multi-agent AI system that follows question question-answering approach to query GPT-4 to extract topics in the Task-Oriented Dialogue (TOD) setting. Ma et al. (2024) proposed an unsupervised dialogue segmentation algorithm to split the dialogue passage into topic-concentrated fragments for dialogue comprehension. These studies do not focus on user preferences for the topics.

Several approaches (Xu et al., 2021; Ren et al., 2022; Wu et al., 2021; Hu et al., 2022) in conversational recommender systems focus on understanding user preferences for items. These approaches interact with users by asking questions about their preference for items and processing user feedback to learn their preferences. To learn user preferences, Xu et al. (2021) uses gating modules, Ren et al. (2022) uses stochastic gradient variational Bayesian (SGVB) estimator, Wu et al. (2021) propose clustering algorithm to cluster users with similar preferences. Hu et al. (2022) employ representation learning. Liu et al. (2024) propose reformulating user preferences as instruction tuning. We do not consider user feedback in this work; therefore, these approaches cannot be applied. More related to our work, Ma et al. (2021b) trains LLMs to learn personalized post representation and construct a general user profile from the user's historical responses. Similarly, Qian et al. (2021) trains LLMs by exploring the conditional relations underneath each post-response pair of the user to learn an implicit user profile from dialogue history.

In this study, we design instructions to query TinyLLama2 (Zhang et al., 2024b) to extract topic and user preferences directly from conversation utterances following the ODD setting. Our proposed EvolvConv does not restrict the topic search space and does not require any additional feedback or external knowledge.

Summarization and Response Generation for Long Conversations: Current models, including large language models (LLMs), struggle to understand long conversation contexts, hindering the generation of responses for long conversations. To overcome this problem, several studies summarize long conversation texts since conversations always contain redundant texts, which make a limited contribution to the overall meaning (Feng et al., 2021).

Some approaches (Zhang et al., 2024a) partition long conversations into fine-grained segments of equal size and apply compression-based language modeling techniques to compress the text. While others follow topic modeling techniques utilizing the topic shifts in the conversation for summarization. Liu et al. (2019) use key points in the paragraph to decode each sub-summary using a Leader-Writer network, Ma et al. (2021a) improved by fixing the type of key points considered and using an MRC-based method to fetch segments. Zou et al. (2021) implicitly modeled topics through token-level salient correspondences. Liu et al. (2021) modeled conversation utterances at the section level to ensure coherence in forming topic segments. Chen and Yang (2020) used multi-view attention to summarize, considering the topic view and stage view. These approaches do not utilize conversation summarization to generate responses.

Different from the above studies, Han et al. (2024) proposes to capture the topic structure of the conversation as a Seq2Seq task and leverage it to guide the generation of the summary. Zhong et al. (2022a) use LLMs as multi-level refiners to extract the most valuable tokens from dialogue history and leverage data from similar users to generate personalized responses.

This study does not summarize the conversation; instead, we extract the conversation topics and user preferences from conversation utterances and utilize them to recommend novel topics to include in the generated responses. Our proposed novel pipeline generates effective long conversations through smooth topic shifting and evolution.

3 Methodology

Given a conversation history $\mathcal{C}$ containing M conversation utterances, our goal is to generate a response $\mathcal{R}$ that best engages the user to continue the conversation. To achieve this goal, we propose EvolvConv[1], a multi-step framework incorporating topic shifting and evolution in response generation. Given the conversation history $\mathcal{C}$, we first extract the discussed explicit and implicit topics and user preferences for each topic from each conversation utterance. While extracting topics, we consider different levels of topic granularity to en-

[1]https://github.com/VT-NLP/EvolvConv

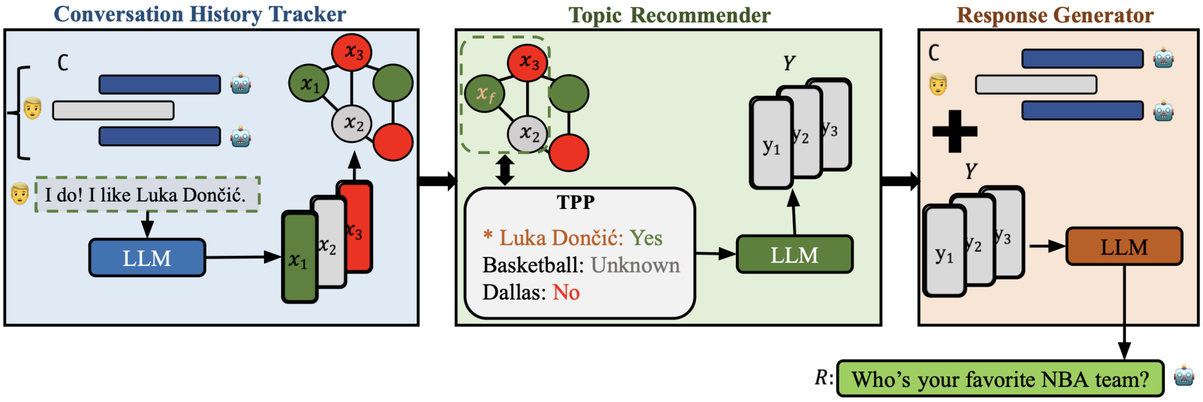


Figure 2: **Overview of EvolvConv.** EvolvConv consists of three modules. Given the conversation history $\mathcal{C}$, the conversation history tracker module extracts the topics and user preferences from $\mathcal{C}$ and stores them as a graph. The user preferences values of *Yes*, *No*, and *Unknown* are represented with green, red, and grey colors, respectively. Then, the topic recommender module retrieves relevant topics at K-hop distance from the current conversation topic (x_f) along with user preferences, which we call Topic Preference Profile (TPP) and utilizes it to recommend topics $\mathcal{Y}$ that decide topic shifting/evolution. The response generator module takes $\mathcal{C}$ and $\mathcal{Y}$ as input and generates a response incorporating recommended topics.

sure a proper understanding of user preferences. For example, certain users may like soccer, while others may like a specific team or player. The user preferences take values in $\{No, Yes, Unknown\}$, representing dislike, like, and no explicit preference for each extracted topic of different granularity. We store the extracted conversation topics and user preferences as a graph. To enable effective topic shifting and evolution and prevent repetition of conversation topics, the topics relevant to current conversation utterances are extracted from the graph and provided as input to the recommender module that recommends topics to include in generated responses. The recommended topics can be novel, aligning with current conversation utterances and extracted user preferences. Finally, the response generator generates responses incorporating recommended topics that align with the conversation history $\mathcal{C}$. Figure 2 provides an overview of EvolvConv.

3.1 Conversation History Tracker

Instead of storing and tracking the entire conversation history, we propose to store and track only the conversation topics and user preferences discussed in the conversation. Topic tracking requires understanding the conversation utterance, extracting important terms, and assigning topics of different granularity to the extracted terms. Furthermore, certain explicit and implicit spans can suggest the user preferences for the extracted terms that need to be extracted. Large language models have been shown to better understand and analyze the input text. Therefore, we propose a training instruction to train generative large language model (LLM) $\mathcal{L}$ to extract topics and user preferences from conversation utterances. The training instruction $\mathcal{I}$ contains task description and conversation utterance $c \in C$ and LLM $\mathcal{L}$ is trained to extract conversation topics along with user preferences for them $\{(x_0, p_0), (x_1, p_1), ..., (x_n, p_n)\}$ from c. x_n represents the extracted topic, and p_n represents the user preference for the topic that can take values in $\{No, Yes, Unknown\}$. We propose a synthesized tracking dataset since none of the existing benchmark datasets are proposed for the task. Section 4.1 discusses the dataset details.

Once LLM $\mathcal{L}$ extracts topics $x_1, x_2, ...x_n$ along with user preferences $p_0, p_1, ..., p_n$, we store them as a graph. The nodes in the graph represent the topics and user preference per topic is stored as node attributes. The topics extracted from each conversation utterance are considered related; therefore, edges connect every pair of extracted topics. We union the nodes and edges for subsequent utterances to update the graph. Due to the union, user preferences need to be updated for common nodes. User preferences can change over time; however, several divergences can be a one-time event. Considering the divergences can result in catastrophic forgetting/overwriting of user preferences. To prevent these issues, we set a threshold λ of consecutive preference updates to update the graph. Figure 3 provides an example from the pro-

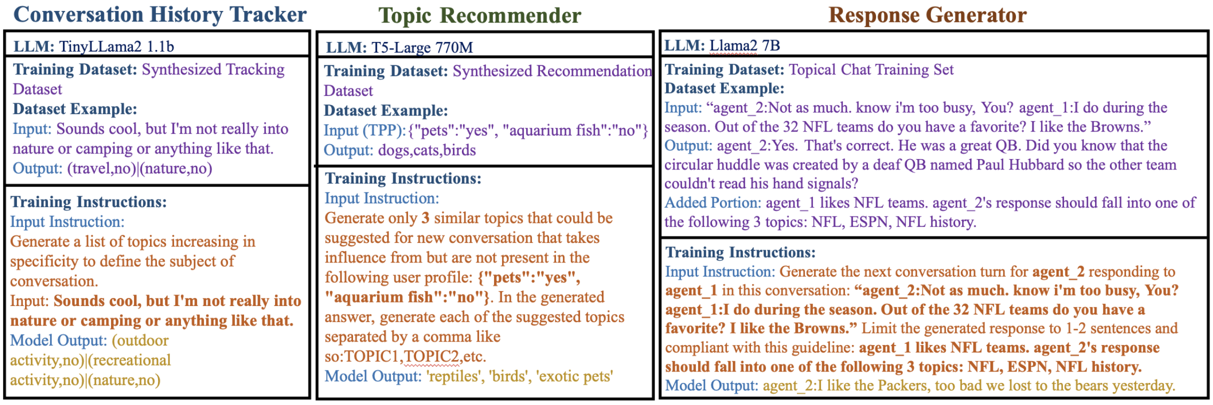


Figure 3: **Dataset and Training Instruction Details.** We propose two datasets, synthesized tracking and recommendation datasets, to train conversation history tracker and topic recommender modules. We use the Topical Chat (Gopalakrishnan et al., 2023) dataset to train our response generator module. We add guidelines (Added Portion) obtained from the recommender module as additional information during training. Training instructions for each module are provided and the variables are highlighted in **bold**.

posed synthesized tracking dataset along with the training instruction $\mathcal{I}$ for LLM $\mathcal{L}$ and generated model output.

3.2 Topic Recommender

Once the graph is constructed, we utilize the graph structure to retrieve potential topics that can be part of the subsequent response. First, we retrieve the topics from the current conversation utterance and randomly choose one of them as the focus node x_f. Since all the related topics to x_f are connected to it through edges, we choose all the nodes, including x_f, and their attributes within K-hop distance. The chosen topics, along with the preferences, form the Topic Preference Profile (TPP). The TPP only contains topics extracted from the conversation history, and using it for response generation results in topic repetition. We propose to train LLM $\mathcal{L}'$ to recommend novel topics aligning with current conversation to enable topic shifting and evolution. LLM $\mathcal{L}'$ takes TPP as input and recommends new topics $\mathcal{Y} = \{y_1, y_2, .., y_z\}$ influenced by TPP without any topics from TPP. Since TPP is only a part of the constructed graph, the recommended topics can be novel or a repetition of topics from the remainder of the graph. The recommended topics $\mathcal{Y}$ are incorporated in the response generator module's response. The recommended novel topics are added to the graph in the next turn, enabling the topic to evolve in subsequent turns. Furthermore, recommended topics $\mathcal{Y}$ can also result in topic shifting since they are influenced by TPP. Similar to the conversation history tracker module, We propose a synthesized recommendation dataset since none of the existing benchmark datasets are proposed for the task. Section 4.1 discusses the dataset format and construction details. Figure 3 provides an example from the proposed synthesized recommendation dataset along with the training instruction $\mathcal{I}'$ for LLM $\mathcal{L}'$ and the generated model output.

3.3 Response Generator

We aim to generate a response that incorporates the recommended topics $\mathcal{Y}$ and aligns with the conversation history ($\mathcal{C}$). Current state-of-the-art generative LLMs are known to generate grammatically accurate responses given the context. Therefore, we use a generative LLM $\mathcal{L}''$ to generate the responses. The input to $\mathcal{L}''$ is conversation history ($\mathcal{C}$), and a guideline $\mathcal{G}$. The guideline $\mathcal{G}$ is constructed from recommended topics ($\mathcal{Y}$) and contains instructions to $\mathcal{L}''$ on what to include in the response, including the information about which user (U) is responding. The guideline, training instruction, and generated model response $\mathcal{R}$ are shown in Figure 3[2]. The conversation history $\mathcal{C}$ helps $\mathcal{L}''$ learn the flow of the conversation; however, $\mathcal{L}''$ does not need the entire conversation history for the purpose. Therefore, if $\mathcal{C}$ becomes lengthy, only recent conversation utterances can be provided as input to $\mathcal{L}''$.

3.4 Proposed Synthesized Datasets

Since our proposed conversation history tracker and topic recommender tasks are novel, the existing benchmark datasets cannot be used. Therefore, we synthesize datasets for both the tasks.

[2]More examples are shown in Section A.4 in Appendix A

3.4.1 Synthesized tracking dataset

The tracking dataset[3] aims to train an LLM to extract topics and corresponding user preferences from conversation utterances. Therefore, the dataset input is a conversation utterance, and the output is the tuple of $\{(x_0, p_0), (x_1, p_1), ..., (x_n, p_n)\}$ of topics and user preferences. Figure 3 shows a sample instance from the synthesized tracking dataset. The dataset is synthesized using GPT-4. The dataset comprises 13,350 conversation utterances from 4,000 conversations covering 44 topics. The utterances reflect typical user interactions observed in popular domains such as movies, food, books, and music and illustrate everyday user conversational trends. We prompt GPT-4 with five annotated in-context examples to generate topic and user preference tuples[4]. The model is asked to generate topics at three levels of granularity. Specifically, the topics are classified as *High-level*, *Middle-level*, and *Low-level* topics. For example, *sports*, *football*, and *Cristiano Ronaldo* are examples of *High-level*, *Middle-level*, and *Low-level* topics, respectively. For each extracted topic, the user preferences are labeled as $\{No, Yes, Unknown\}$, representing dislike, like, and no explicit preference.

3.4.2 Synthesized recommendation dataset

The recommendation dataset[5] aims to train an LLM to recommend topics similar to the input TPP. Therefore, the input of the recommendation dataset is a TPP and the output is comma separated recommended topics. Figure 3 shows a sample instance from the synthesized recommendation dataset. We use GPT-4 to synthesize the dataset. The process of dataset synthesis is discussed in Appendix A. The synthesized dataset contains 10,307 instances of TPP and recommendation topic pairs. The TPPs in the dataset cover 1,403 unique topics, with each TPP containing an average of 2.215 topics, with the maximum and minimum number of topics being 10 and 1, respectively. Similarly, the recommendation topics cover 5,666 unique topics (1,126 of these topics overlap with the TPP), with an average of 3.16, a maximum of 15, and a minimum of 1.

[3]https://huggingface.co/datasets/TrevorAshby/EvolvConv-Track

[4]GPT-4 template along with five annotated in-context examples are discussed in Section A.4 in Appendix A

[5]https://huggingface.co/datasets/TrevorAshby/EvolvConv-Recommend

3.5 LLM Model Selection

We selected an instruction-tuned 1.1 billion parameter LLaMA2 model for the topic tracking task due to its ability to handle nuanced and complex instructions while also reducing the inference and computational complexity. The decision to use T5 Large for topic recommendation follows the need to further reduce parameter numbers and computational demands. For the core task of response generation, we selected an 8 billion parameter LLaMA2 model that benefits from extensive instruction tuning, allowing us to incorporate relevant contextual information directly into the input prompts and capable of complex queries and generate detailed, contextually appropriate responses. While these LLMs were selected in this work, our pipeline is foundation model agnostic, and any LLM can be hot-swapped for other models depending on use case.

4 Experiments

To test the topic shifting and evolving capabilities of EVOLVCONV, we evaluated its performance on several benchmark datasets. We provide additional evaluation studies in Appendix A.

4.1 Datasets

We first discuss the training datasets for EVOLVCONV and then provide information about the datasets used for testing. The dataset statistics are provided in Appendix A in Table 4.

4.1.1 Training datasets

We use synthesized tracking and recommendation datasets discussed in Section 4.1 to train the conversation history tracker and topic recommender modules of EVOLVCONV. To train the response generator module, we use the train set of Amazon's Topical Chat (Gopalakrishnan et al., 2023). In addition to the dataset input, we add the guideline generated using the recommended topics as additional input. Figure 3 shows an example instance of the topical chat dataset along with the generated guideline and the instruction for model training.

4.1.2 Testing datasets

We compare the responses of EVOLVCONV with those of the baselines for three benchmark datasets. We use validation and test sets of Amazon's Topical Chat (Gopalakrishnan et al., 2023), test set of TIAGE (Xie et al., 2021) (topic-shift-aware dialogue) benchmark, and test set of MultiWOZ2.1 (Budzianowski et al., 2018; Ramadan et al., 2018; Eric et al., 2019; Zang et al., 2020) (Multi-domain Wizard of Oz V2.1) datasets for testing.

4.2 Baselines

We compare EvolvConv with three baselines that follow different settings. **(1) Zero-shot setting:** In this setting, we use pre-trained 7 billion parameter LLama2 (Touvron et al., 2023), which we call **L2-Zero**. The input to the model is the conversation history, and the output is the response conversation utterance. **(2) Fine-tuned setting:** In this setting, we fine-tune the conversational AI model Vicuna on the topical chat dataset, which we call **Vic-Fine**. Again, the input to the model is the conversation history, and the output is the response conversation utterance. **(3) Topic-aware response generator:** We use OTTers (Sevegnani et al., 2021) which generates responses from topical one-hop transitions. The input to OTTers is the previous (c_{m-1}) and next (c_{m+1}) conversation utterance and generates current (c_m) conversation utterance that bridges c_{m-1} and c_{m+1}. OTTers assumes we have some idea about the future (c_{m+1}), which differs from our setting. However, we provide the next (c_{m+1}) conversation utterance as input to compare with the baseline.

4.3 Evaluation Metrics

Our goal is to evaluate all the nuanced aspects of the conversation to test the practical viability of EvolvConv. Therefore, we use an *un-referenced evaluation metric* **UniEval** (Zhong et al., 2022b) that tests the responses for six aspects such as *Naturalness*, *Coherence*, *Engagingness*, *Groundedness*, *Understandability*, and *Overall*[6]. Furthermore, to test the practical usability of EvolvConv, we conduct a user survey, where users rate the response of each system for a given conversation. We provide user survey template in Appendix A in Figure 5.

4.4 Experimental Settings

To ensure practical usability, we use models with fewer parameters to train modules of EvolvConv. For the topic tracking module, we train the 1.1b parameter LLama2 model (Zhang et al., 2024b). The model is trained for 1 epoch with a learning rate of $1e-5$ and batch size of 32. For the recommendation module, we train the 744M parameter T5 model (Raffel et al., 2023). The model is trained on 90% of our proposed recommendation dataset for 5 epochs with a learning rate of $1e-4$ and batch size of 64. For the realistic response generator module, we train 7b parameter LLama2 model (Touvron et al., 2023) for 6 epochs with learning rate of $5e-4$ and batch size of 1.

[6]The results are discussed in Appendix A

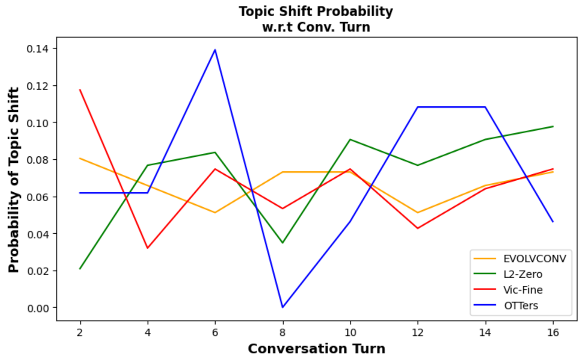


Figure 4: Comparison of Topic Shifting probability. The plot shows the topic-shifting probability of each model at a given turn.

4.5 Results and Discussion

This section discusses the experiments conducted to test the conversational capabilities of EvolvConv compared to the baselines.

Table 1: Results for topic evolution capabilities of models. DC represents disconnected components in the graph.

Baseline	Avg. DC	Avg. DC Nodes	Avg. DC Edges	Avg. Nodes
EvolvConv	5.0	3.5	3.167	15.667
L2-Zero	3.	5.6	5.6	13.0
Vic-Fine	9.333	1.456	0.522	13.667

Table 2: User Survey Ranking Results. Row totals are not identical due to the participants ability to rank up to 2 responses the same rank.

Baseline	Rank1	Rank2	Rank3	Rank4
EvolvConv	11	7	3	9
L2-Zero	4	3	6	17
Vi-Fine	8	8	12	2
Human Resp.	13	13	4	0

4.5.1 Topic-shifting capability of models

We conduct experiments to validate if EvolvConv shifts topics smoothly or abruptly compared to the baselines. "Quality" of topic shift is an abstract metric to evaluate; therefore, we compare the probability of topic shift at each turn in the conversation. Since automatic evaluation is not possible, we conduct manual evaluation. Since only the

TIAGE dataset has human-annotated topic shifts, we randomly select 10 conversation instances of size 16 from it for the experiment.

For each conversation, we incrementally generate responses for each turn. Specifically, we generate responses for turns 1, 2,..., and 16 using each model and manually evaluate the probability of topic shift at each turn. Figure 4 shows the experiment results. From the results, we can observe that the probability of topic-shifting is stable for EVOLVCONV compared to the baselines. Specifically, the topic shifting probability of EVOLVCONV is between $5\% - 8\%$ for all turns, whereas the probability ranges between $2\% - 9\%$, $3\% - 12\%$, and $0\% - 14\%$ for L2-Zero, Vic-Fine, and OTTers, respectively. The probability shifting pattern demonstrates that EVOLVCONV can smoothly shift topics throughout the conversation without any abrupt shifts. Furthermore, the initial drop in probability from turn 0 to 6 shows that EVOLVCONV can better handle the introductory statements in a conversation, allowing early topics proper time to develop before shifting the topic.

4.5.2 Topic evolution capabilities of models

We conduct experiments to test the topic evolution capabilities of EVOLVCONV compared to the baselines. If EVOLVCONV is used in a real-life setting, it should converse in a chatbot style with the user. To align with real-life scenarios, we experiment with a human participant. We ask the participant to chat with EVOLVCONV and the baselines on a pre-defined topic and for a pre-defined number of turns. We obtain topics and number of turns for the experiment from the test sets of Topical Chat, TIAGE, and MultiWOZ2.1 datasets. The topics are randomly sampled from the topics discussed in the datasets, and the number of turns is set to the average number of turns in the dataset. We resample if random sampling results in an overlap in topics between datasets. For the Topical Chat dataset, the sampled topics are *Football, Radio, Basketball*, and the number of turns is set to 22. For the TIAGE dataset, the sampled topics are *Weather Seasons, Fishing, Education*, and the number of turns is set to 16. The sampled topics for the MultiWOZ2.1 dataset are *Reservation, Restaurant, Hotel*, and the number of turns is set to 14. Overall, the participant converses with each system nine times.

For each method, we extract the topics from the generated conversation using our conversation history tracker module and construct the graph. We compare the statistics of the constructed graphs. Table 1 shows the comparison statistics. Comparing the average number of nodes in the graphs, we can observe that EVOLVCONV can generate more topics than the baselines. The disconnected components in the graph represent related topics. Looking at the disconnected components in the graphs, we observe that Vic-Fine generates the largest number of disconnected components; however, the average number of nodes and edges in the disconnected components is few, which shows that Vic-Fine abruptly evolves the topics without giving sufficient time to develop the conversation. Users may not enjoy the conversation if the topic evolves abruptly. Looking at L2-Zero, we observe that it generates the fewest disconnected components with the highest number of nodes and edges. The results show that L2-Zero does not evolve topics and repeats the topics discussed in the conversation. Again, users may not enjoy a conversation where the topics repeat. Looking at EVOLVCONV, we observe that it generates a good number of disconnected components with sufficient nodes and edges for topic development. Users would enjoy a conversation that develops smoothly with sufficient time for each topic discussed. Overall, we can conclude that EVOLVCONV outperforms the baselines with a significant margin for topic evolution capabilities.

4.5.3 Effect of conversation history size

We consider the validation split of Topical Chat and test splits of Topical Chat, TIAGE, and MultiWOZ2.1 datasets for the experiments. For each dataset split, we select conversations of sizes 3, 12, and 20. For the performance comparison, we compute the 5 aspects *Naturalness*, *Coherence*, *Engagingness*, *Understandability*, and *Overall* of UniEval score. In Table 3, we report the loss in the Overall UniEval score as the conversation size increases. Table 7 in Appendix A reports individual scores. From the results in Table 3, we can observe that loss in Overall UniEval score is minimal for EVOLVCONV compared to the baselines for three out of four datasets for *Sml=3, Lrg=12* and *Sml=3, Lrg=20* settings and comes second for three out of four datasets for *Sml=12, Lrg=20* setting. The results confirm that the proposed solution can limit performance degradation as conversation history size increases. For BLEU and ROUGE scores see Table 8 in Appendix A.

Table 3: The % UniEval 'Overall' retention score (UniEval) as the size of conversation history increases. The values in the table represent the loss in UniEval score as the conversation history size increases from **Sml** to **Lrg**. L2-Zero represents LLama2 baseline that follows zero-shot setting, Vic-Fine represents fine-tuned Vicuna baseline. The results of best performing framework are highlighted in **bold**.

Dataset	Split	Sml=3, Lrg=12			Sml=3, Lrg=20			Sml=12, Lrg=20		
		L2-Zero	Vic-Fine	EVOLVCONV	L2-Zero	Vic-Fine	EVOLVCONV	L2-Zero	Vic-Fine	EVOLVCONV
Topical Chat	Valid	19	12.4	**9.8**	24.3	23.2	**22.2**	**6.6**	12.4	13.7
Topical Chat	Test	18.9	15.7	**14.5**	23.2	25.8	**20.7**	**5.2**	12.1	7.3
TIAGE	Test	20.1	15.2	**5.8**	23.5	9.9	**5.8**	4.4	**-0.1**	0.1
MultiWOZ2.1	Test	**11.3**	27.2	32.6	**16.1**	46.2	45.4	**5.4**	26.1	19.0

4.5.4 User preference modeling for long conversations

EVOLVCONV can model user preferences for long conversations better than the baselines if humans prefer its generated responses over the baselines. To test the human preference of EVOLVCONV, we conduct a user survey with 6 participants. In the survey, the participants are asked to rank the responses produced by EVOLVCONV, L2-Zero, Vic-Fine, and humans on a scale of 1-4 (1 being the highest preference and 4 being the lowest preference) based on their judgment of how well the response captures user preferences and fits into the conversation. The user survey format and an example are provided in Appendix A Figure 5. For the experiment, we randomly selected 6 long conversations, 2 from the test set of Topic Chat of lengths 12, and 20, 2 from TIAGE of lengths 12, and 15, and 2 from MultiWOZ2.1 of lengths 9, and 11. The participants rank the responses for each turn in the conversation. Table 2 shows the user survey rankings provided by participants. From the table, we can observe that EVOLVCONV is preferred by the participants for long conversations compared to the baselines and the responses generated by EVOLVCONV are comparable to human generated responses. The results further confirm that EVOLVCONV can overcome the issue of performance degradation for long conversations faced by baselines.

5 Conclusion

This work proposes EVOLVCONV, a multi-step model that utilizes dynamic topic tracking and recommendation to perform topic shifting and evolution for effective long conversation generation. Instead of storing the entire conversation history, EVOLVCONV only stores topics and corresponding user preferences as a graph. Then, the graph is utilized to retrieve TPP, which form the input to the recommender module that is responsible for topic shifting and evolution in the responses. Finally, the response generator generates responses incorporating recommended topics and aligning with the conversation flow. Through extensive experiments, we demonstrate the topic-shifting and evolving capabilities of EVOLVCONV for long conversations, including the ability to model user preferences effectively.

6 Ethics Statement

We comply with the ACL Code of Ethics. For the experiments, we use large language models that follow ethical considerations. Our user survey experiments are conducted on very few samples, and we report the template of the user survey in Figure 5 of Appendix A. The participants chosen for the survey are selected at random, and they do not have any affiliation with our lab or the university. We do not collect any personally identifiable information; the only information we collect is the participant's response to the survey. Participants are not provided with any monetary benefit for the survey. We provide further details about the steps followed for an unbiased survey in the *Limitations* section A.1 in Appendix A.

References

Paige H Adams and Craig H Martell. 2008. Topic detection and extraction in chat. In *2008 IEEE international conference on Semantic computing*, pages 581–588. IEEE.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, 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 (EMNLP)*.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD Explor. Newsl.*, 19(2):25–35.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4106–4118, Online. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, and Ting Liu. 2021. Language model as an annotator: Exploring DialoGPT for dialogue summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1479–1491, Online. Association for Computational Linguistics.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. 2023. Topical-chat: Towards knowledge-grounded open-domain conversations. *Preprint*, arXiv:2308.11995.

Qinyu Han, Zhihao Yang, Hongfei Lin, and Tian Qin. 2024. Let topic flow: A unified topic-guided segment-wise dialogue summarization framework. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2021–2032.

Chenhao Hu, Shuhua Huang, Yansen Zhang, and Yubao Liu. 2022. Learning to infer user implicit preference in conversational recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 256–266.

Eric Klinger. 2014. The contents of thoughts: Interference as the downside of adaptive normal mechanisms in thought flow. In *Cognitive interference*, pages 3–24. Routledge.

Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019. Automatic dialogue summary generation for customer service. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 1957–1965, New York, NY, USA. Association for Computing Machinery.

Junpeng Liu, Yanyan Zou, Hainan Zhang, Hongshen Chen, Zhuoye Ding, Caixia Yuan, and Xiaojie Wang. 2021. Topic-aware contrastive learning for abstractive dialogue summarization. *Preprint*, arXiv:2109.04994.

Minqian Liu, Ying Shen, Zhiyang Xu, Yixin Cao, Eunah Cho, Vaibhav Kumar, Reza Ghanadan, and Lifu Huang. 2024. X-eval: Generalizable multi-aspect text evaluation via augmented instruction tuning with auxiliary evaluation aspects. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8560–8579, Mexico City, Mexico. Association for Computational Linguistics.

Bing Ma, Cao Liu, Jingyu Wang, Shujie Hu, Fan Yang, Xunliang Cai, Guanglu Wan, Jiansong Chen, and Jianxin Liao. 2021a. Distant supervision based machine reading comprehension for extractive summarization in customer service. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1895–1899, New York, NY, USA. Association for Computing Machinery.

Xinbei Ma, Yi Xu, Hai Zhao, and Zhuosheng Zhang. 2024. Multi-turn dialogue comprehension from a topic-aware perspective. *Neurocomputing*, 578:127385.

Zhengyi Ma, Zhicheng Dou, Yutao Zhu, Hanxun Zhong, and Ji-Rong Wen. 2021b. One chatbot per person: Creating personalized chatbots based on implicit user profiles. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 555–564, New York, NY, USA. Association for Computing Machinery.

Jihyun Park, Dimitrios Kotzias, Patty Kuo, Robert L Logan IV, Kritzia Merced, Sameer Singh, Michael Tanana, Efi Karra Taniskidou, Jennifer Elston Lafata, David C Atkins, Ming Tai-Seale, Zac E Imel, and Padhraic Smyth. 2019. Detecting conversation topics in primary care office visits from transcripts of patient-provider interactions. *Journal of the American Medical Informatics Association*, 26(12):1493–1504.

Hongjin Qian, Zhicheng Dou, Yutao Zhu, Yueyuan Ma, and Ji-Rong Wen. 2021. Learning implicit user profile for personalized retrieval-based chatbot. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM '21. ACM.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 432–437.

Zhaochun Ren, Zhi Tian, Dongdong Li, Pengjie Ren, Liu Yang, Xin Xin, Huasheng Liang, Maarten de Rijke, and Zhumin Chen. 2022. Variational reasoning about user preferences for conversational recommendation. In *Proceedings of the 45th International ACM*

SIGIR Conference on Research and Development in Information Retrieval, pages 165–175.

Karin Sevegnani, David M. Howcroft, Ioannis Konstas, and Verena Rieser. 2021. Otters: One-turn topic transitions for open-domain dialogue. *Preprint*, arXiv:2105.13710.

Virginia Tech. 2023. Hokiebot: Towards personalized open-domain chatbot with long-term dialogue management and customizable automatic evaluation. In *Alexa Prize SocialBot Grand Challenge 5 Proceedings*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Junda Wu, Canzhe Zhao, Tong Yu, Jingyang Li, and Shuai Li. 2021. Clustering of conversational bandits for user preference learning and elicitation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2129–2139.

Yuxia Wu, Tianhao Dai, Zhedong Zheng, and Lizi Liao. 2024. Active discovering new slots for task-oriented conversation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2062–2072.

Huiyuan Xie, Zhenghao Liu, Chenyan Xiong, Zhiyuan Liu, and Ann Copestake. 2021. TIAGE: A benchmark for topic-shift aware dialog modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1684–1690, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kerui Xu, Jingxuan Yang, Jun Xu, Sheng Gao, Jun Guo, and Ji-Rong Wen. 2021. Adapting user preference to online feedback in multi-round conversational recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 364–372.

Dejian Yu and Bo Xiang. 2023. Discovering topics and trends in the field of artificial intelligence: Using lda topic modeling. *Expert Systems with Applications*, 225:120114.

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, ACL 2020*, pages 109–117.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2024a. Compressing lengthy context with ultragist. *arXiv preprint arXiv:2405.16635*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024b. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. *Preprint*, arXiv:1911.00536.

Hanxun Zhong, Zhicheng Dou, Yutao Zhu, Hongjin Qian, and Ji-Rong Wen. 2022a. Less is more: Learning to refine dialogue history for personalized dialogue generation. *Preprint*, arXiv:2204.08128.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022b. Towards a unified multi-dimensional evaluator for text generation. *Preprint*, arXiv:2210.07197.

Yicheng Zou, Lujun Zhao, Yangyang Kang, Jun Lin, Minlong Peng, Zhuoren Jiang, Changlong Sun, Qi Zhang, Xuanjing Huang, and Xiaozhong Liu. 2021. Topic-oriented spoken dialogue summarization for customer service with saliency-aware topic modeling. *Preprint*, arXiv:2012.07311.

A Appendix

In this section, we discuss the limitations of EvolvConv in Section A.1. We conduct ablation studies to test the effect of model parameter size and conversation history size on the performance of EvolvConv. The details are discussed in Section A.2. Furthermore, we provide the details of dataset statistics used for experiments in Table 4, the template for user survey in Figure 5, example outputs of conversation history tracker, topic recommender and response generator modules in Table 9 and 10, and provide the input template of GPT-4 model used to synthesize tracking dataset in Figure 6 and example outputs in Figure 7.

A.1 Limitations

The proposed EvolvConv is a pipeline-based framework prone to error propagation. Furthermore, due to multiple modules, EvolvConv required higher training time than single-step frameworks. Furthermore, since EvolvConv uses several foundation models, the limitations of these models are also applied to EvolvConv. We made every possible effort to ensure that the human annotators chosen for evaluation are unbiased. Furthermore, the annotators are not provided with any extra information apart from the user survey template shown in Figure 5. However, personal human preference may guide user decisions. Since our goal is for practical applicability of EvolvConv, we believe personal human preference does not affect our observations.

A.2 Ablation Studies

A.3 Effect of model parameter sizes

For practical usability, we use models with fewer parameters in EvolvConv. In these experiments, we compare our chosen models for conversation history tracker and topic recommender modules with higher parameter models to analyze the effect of model parameter size on performance. Specifically, we use LLama2 7b (Touvron et al., 2023) model for both modules. For training LLama2 7b for both modules, the number of epochs is set to 6, the learning rate to $5e-4$, and the batch size is set to 1. For conversation history tracker and topic recommender modules, the models are trained on 90% and tested on the remaining 10% of synthesized tracking and recommendation datasets, respectively. Instead of treating the tasks as strict classification tasks, we evaluate the cosine similarities between the predictions x^p and ground truth x^*. We calculate Precision, Recall, and F1-scores of computed cosine similarities as follows:

$$cos(x^p, x^*) = 1 - \frac{x^p \cdot x^*}{||x^p|| * ||x^*||},$$

$$Prec = \frac{1}{t}\sum_{i=1}^{t} max[cos(x_i^p, x_0^*), ..., cos(x_i^p, x_z^*)],$$

$$Rec = \frac{1}{z}\sum_{j=1}^{z} max[cos(x_j^*, x_0^p), ..., cos(x_j^*, x_t^p)],$$

$$F1 = \frac{2 * Prec * Rec}{Prec + Rec}.$$

Here, t and z represent the cardinality of predicted and ground truth sets. The results for the conversation history tracker module are shown in Table 5, and the results for the topic recommender module are shown in Table 6. From the results, we can observe that models with a larger number of parameters do not improve the performance of the models. In fact, models with fewer parameters achieve significantly better performance. Our analysis revealed that the higher parameter model tends to overlook the high-level general topics and tends to extract fine-grained topics, resulting in overcomplication for simpler cases and a drop in performance.

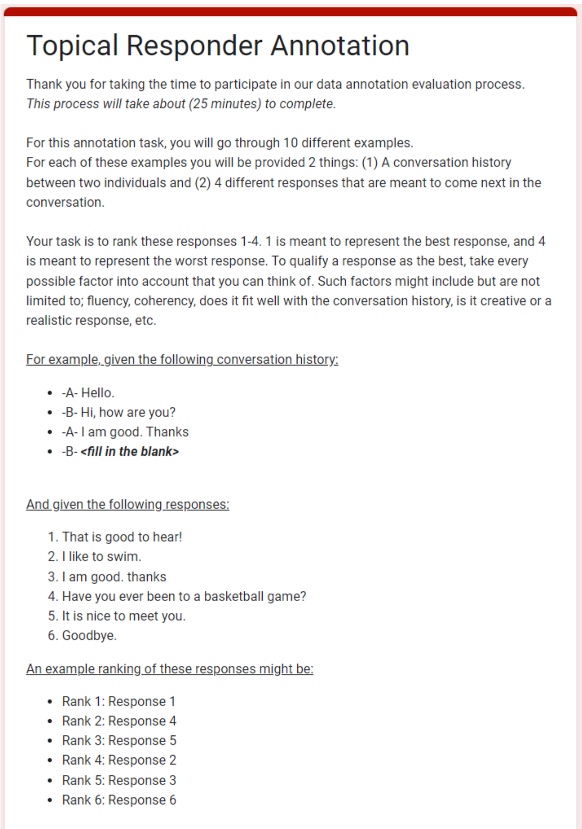
A.4 Examples

We provide five example outputs of the end-to-end flow of proposed EvolvConv in Table 9 and 10. Specifically, we provide the input conversation history, the output of the conversation history tracker module, the generated topic preference profile (TPP), the output of the topic recommender module, the guideline generated from the recommended topics, and the final response generated by the response generator module.

We also provide the information about the input template used for the GPT-4 model along with the five in-context examples used to synthesize the tracking dataset in Figure 6 and the examples of the synthesized tracking dataset in Figure 7.

Table 4: Dataset Statistics

Dataset	Split	# Conversations $\mathcal{C}$	# of $\mathcal{C}$ snippets
Topical Chat (Gopalakrishnan et al., 2023)	Train	8,628	188,378
HOKIEBOT (Tech, 2023)	Full	4,000	13,350
Topical Chat (Gopalakrishnan et al., 2023)	Valid	539	11,681
Topical Chat (Gopalakrishnan et al., 2023)	Test	539	11,760
TIAGE (Xie et al., 2021)	Test	500	7861
MultiWOZ2.1 (Budzianowski et al., 2018; Ramadan et al., 2018; Eric et al., 2019; Zang et al., 2020)	Test	1000	13,460



Topical Responder Annotation

Thank you for taking the time to participate in our data annotation evaluation process. *This process will take about (25 minutes) to complete.*

For this annotation task, you will go through 10 different examples.
For each of these examples you will be provided 2 things: (1) A conversation history between two individuals and (2) 4 different responses that are meant to come next in the conversation.

Your task is to rank these responses 1-4. 1 is meant to represent the best response, and 4 is meant to represent the worst response. To qualify a response as the best, take every possible factor into account that you can think of. Such factors might include but are not limited to; fluency, coherency, does it fit well with the conversation history, is it creative or a realistic response, etc.

For example, given the following conversation history:

- -A- Hello.
- -B- Hi, how are you?
- -A- I am good. Thanks
- -B- ***<fill in the blank>***

And given the following responses:

1. That is good to hear!
2. I like to swim.
3. I am good. thanks
4. Have you ever been to a basketball game?
5. It is nice to meet you.
6. Goodbye.

An example ranking of these responses might be:

- Rank 1: Response 1
- Rank 2: Response 4
- Rank 3: Response 5
- Rank 4: Response 2
- Rank 5: Response 3
- Rank 6: Response 6

Figure 5: User Survey Template

Table 5: Performance comparison between different parameter models for conversation history tracker module. The results of best performing models are highlighted in **bold**.

Model	Output	Prec.	Rec.	F1
LLama2 (1.1b)	Topic	**77.6**	**74.6**	**76.1**
LLama2 (7b)	Topic	71.2	70.3	70.7
LLama2 (1.1b)	Preference	**92.7**	**89.3**	**90.9**
LLama2 (7b)	Preference	89.9	89.1	89.5

Table 6: Performance comparison between different parameter models for topic recommender module. The results compare the recommended topics. The results of best performing models are highlighted in **bold**.

Model	Prec.	Rec.	F1
T5 (744M)	**67.3**	65.2	**66.2**
LLama2 (7b)	65.4	65.2	65.3

Here are 5 examples of a conversation containing 3 pieces: the conversation history, user topic preferences, and the guidelines for a chat assistant. Each of these are separated by the "|" token.

(1) B:Do you like eating food? A:I love eating most kinds of food. B:What is something that you do not like? A:I do not like mexican food. |{"high-level": {"topic": "food", "if_interest": "yes"}, "middle-level": {"topic": "Mexican food", "if_interest": "no"}} | The user is interested in talking about food. They do not like Mexican food, so talk about another type of food.
(2) B:What do you like to do? A:I like listening to rock n roll music. I really like The Beatles and Elvis Presely. | {"high-level": {"topic": "music", "if_interest": "yes"}, "middle-level": {"topic": "rock n roll", "if_interest": "yes"}} {"high-level": {"topic": "music", "if_interest": "yes"}, "middle-level": {"topic": "bands/artists", "if_interest": "yes"}, "low-level": {"topic": "The Beatles/Elvis Presley", "if_interest": "yes"}} | The user likes to listen to music. They like the rock n roll genre. They like the band 'The Beatles' and the artist 'Elvis Presely'. Tell them about other rock n roll artists similar to 'The Beatles' and 'Elvis Presely'.
(3) B:What is a hobby that you like? A:I like reading. I like reading fantasy books, but I do not like 'Dune'. | {"high-level": {"topic": "reading", "if_interest": "yes"}} {"high-level": {"topic": "reading", "if_interest": "yes"}, "middle-level": {"topic": "genre", "if_interest": "yes"}, "low-level": {"topic": "fantasy", "if_interest": "yes"}} {"high-level": {"topic": "reading", "if_interest": "yes"}, "middle-level": {"topic": "book", "if_interest": "unknow"}, "low-level": {"topic": "Dune", "if_interest": "no"}} | The user likes to reed books. They specifically like to read fantasy books. They are not interested in reading the book 'Dune'. Talk to them about any other potential books that they like reading.
(4) A:I do not like sushi. B:What kind of food do you like? A:I like Italian and Mexican cuisine. B:What Italian and Mexican dishes are your favorite? A:Lasagna, spaghetti bolognese, tacos, and burritos. | {"high-level": {"topic": "food", "if_interest": "no"}, "low-level": {"topic": "sushi", "if_interest": "no"}} {"high-level": {"topic": "food", "if_interest": "yes"}, "middle-level": {"topic": "cuisine", "if_interest": "yes"}, "low-level": {"topic": ["Italian", "Mexican"], "if_interest": "yes"}} {"high-level": {"topic": "food", "if_interest": "yes"}, "middle-level": {"topic": "Italian cuisine", "if_interest": "yes"}, "low-level": {"topic": ["lasagna", "spaghetti bolognese"], "if_interest": "yes"}} {"high-level": {"topic": "food", "if_interest": "yes"}, "middle-level": {"topic": "Mexican cuisine", "if_interest": "yes"}, "low-level": {"topic": ["tacos", "burritos"], "if_interest": "yes"}} | The user does not like the food sushi. However, they like Italian and Mexican cuisine. They specifically like lasagna, spaghetti bolognese, tacos, and burritos. Ask them about some other Italian or Mexican cuisine dishes that they like or that you think they would like to try.
(5) A:TV series are not my favorite, but I do like comedy. B:Do you like Game of Thrones? A: No. B:What comedies do you like? A:I like the office. My favorite moment from it is the dinner party episode. B:What is another comedy that you like? A:I also really enjoy Friends. | {"high-level": {"topic": "TV series", "if_interest": "yes"}, "middle-level": {"topic": "Game of Thrones", "if_interest": "no"}} {"high-level": {"topic": "TV series", "if_interest": "no"}} {"high-level": {"topic": "TV series", "if_interest": "yes"}, "middle-level": {"topic": "comedies", "if_interest": "yes"}, "low-level": {"topic": "The Office", "if_interest": "yes"}} {"high-level": {"topic": "TV series", "if_interest": "yes"}, "middle-level": {"topic": "favorite moment", "if_interest": "yes"}, "low-level": {"topic": "Dinner Party episode", "if_interest": "yes"}} {"high-level": {"topic": "TV series", "if_interest": "yes"}, "middle-level": {"topic": "comedies", "if_interest": "yes"}, "low-level": {"topic": "Friends", "if_interest": "yes"}} | The user is generally not interested in TV series. They specifically do not like 'Game of Thrones'. They are however, interested in the comedy 'The Office'. Their favorite moment from the series is the 'Dinner Party' episode. Another TV series they like is 'Friends'. Ask the user why they don't like most TV series other than comedies.

Generate 20 new and unique examples similar to the provided 5. Include all 3 pieces: the conversation history, the topic preferences, and the guidelines for the conversation. Make each generated example different from each other, but make sure to follow the format seen in the previous 5 examples. Make sure that the examples alternate how many preferences are present in each generation.

Figure 6: GPT-4 template along with in-context examples to synthesize tracking dataset.

Example 1:
Conversation:
...... A: Do you like sports? B: Yes, I do. I particularly enjoy basketball.
Topics: {“high-level”:{“topic”:“sports”,“if_interest”:“yes”},“middle-level {“topic”: “basketball”, “if_interest”: “yes”}}
Guidance: The user likes sports and basketball. Talk to them about their favorite basketball teams or players.

Example 2:
Conversation:
...... A:I like playing video games. B:What type of video games do you enjoy? A:I like playing RPGs and action/adventure games.
Topics: {“high-level”: {“topic”: “video games”, “if_interest”: “yes”}, “middle-level”: {“topic”: “genre”, “if_interest”: “yes”}, “low-level”: {“topic”: [“RPG”, “action/adventure”], “if_interest”: “yes”}}
Guidance: The user enjoys playing video games in the RPG and action/adventure genres. Ask about their favorite game or suggest a new one they may enjoy.

Example 3:
Conversation:
...... A: I’m not interested in politics. B: What other current events are you interested in? A: I enjoy following the stock market.
Topics: {“high-level”: {“topic”: “current events”, “if_interest”: “yes”}, “middle-level”: {“topic”: “politics”, “if_interest”: “no”}} {“high-level”: {“topic”: “current events”, “if_interest”: “yes”}, “middle-level”: {“topic”: “finance”, “if_interest”: “yes”}, “low-level”: {“topic”: “stock market”, “if_interest”: “yes”}}
Guidance: The user is not interested in politics, but they like following the stock market. Ask them about their knowledge of finance and suggest similar topics they might want to know about.

Figure 7: Example of synthesized tracking dataset.

Table 7: The change in UniEval Naturalness, Coherence, Engagingness, Understandability scores as the size of conversation history increases. The values in the table represent the loss in UniEval scores as the conversation history size increases from **Sml** to **Lrg**. L2-Zero represents LLama2 baseline that follows zero-shot setting, Vic-Fine represents fine-tuned Vicuna baseline. The results of best performing framework are highlighted in **bold**.

Dataset	Split	Aspect	Sml=3, Lrg=12			Sml=3, Lrg=20			Sml=12, Lrg=20		
			L2-Zero	Vic-Fine	EVOLVCONV	L2-Zero	Vic-Fine	EVOLVCONV	L2-Zero	Vic-Fine	EVOLVCONV
Topical Chat	Valid	Naturalness	**0.019**	0.000	0.010	**0.022**	0.000	-0.013	**0.003**	0.000	-0.024
		Coherence	0.000	-0.050	**0.111**	**0.000**	-0.049	-0.032	0.000	**0.001**	-0.143
		Engagingness	0.000	0.047	**0.069**	0.000	-0.106	**0.008**	**0.000**	-0.153	-0.062
		Understandability	**0.019**	0.000	0.010	-0.818	**0.000**	-0.016	-0.837	**0.000**	-0.026
Topical Chat	Test	Naturalness	**0.021**	0.000	0.000	**0.012**	0.000	-0.009	-0.009	**0.000**	-0.010
		Coherence	**0.000**	-0.101	-0.056	**0.000**	-0.165	-0.137	**0.000**	-0.065	-0.081
		Engagingness	0.000	-0.023	**0.004**	0.000	-0.189	**0.006**	**0.000**	-0.166	0.001
		Understandability	**0.022**	0.000	0.002	**0.010**	0.000	-0.012	-0.012	**0.000**	-0.013
TIAGE	Test	Naturalness	**0.048**	0.000	0.012	**0.024**	0.000	-0.005	-0.024	**0.000**	-0.017
		Coherence	0.000	0.024	**0.283**	0.000	0.233	**0.398**	0.000	**0.209**	0.115
		Engagingness	0.000	0.052	**0.256**	0.000	0.151	**0.309**	0.000	**0.099**	0.053
		Understandability	**0.050**	0.000	0.011	**0.024**	0.000	-0.005	-0.026	**0.000**	-0.016
MultiWOZ2.1	Test	Naturalness	**0.025**	0.000	-0.010	**0.014**	0.000	-0.017	**0.014**	0.000	-0.007
		Coherence	**0.000**	-0.287	-0.301	**0.000**	-0.519	-0.462	**0.000**	-0.231	-0.161
		Engagingness	**0.000**	-0.193	-0.243	**0.000**	-0.407	-0.371	**0.000**	-0.213	-0.128
		Understandability	**0.028**	0.000	-0.012	**0.021**	0.000	-0.021	**0.021**	0.000	-0.009

Table 8: The value of referenced evaluation metrics BLEU and ROUGE for different window sizes. L2-Zero represents LLama2 baseline that follows zero-shot setting, Vic-Fine represents fine-tuned Vicuna baseline. The results of best performing framework are highlighted in **bold**.

Dataset	Split	Window	BLEU			ROUGE		
			L2-Zero	Vic-Fine	EVOLVCONV	L2-Zero	Vic-Fine	EVOLVCONV
Topical Chat	Valid	3	0.1029	0.1468	**0.1547**	0.098	0.1111	**0.1122**
		12	0.0994	**0.1438**	0.1422	0.0973	**0.1131**	0.1089
		20	0.0555	**0.1293**	0.1091	0.0752	**0.1261**	0.1022
Topical Chat	Test	3	0.1001	**0.1613**	0.1554	0.0996	**0.1193**	0.1156
		12	0.1048	**0.1508**	0.1475	0.1014	**0.1192**	0.115
		20	0.057	**0.1319**	0.1128	0.0812	**0.1339**	0.1111
TIAGE	Test	3	0.029	0.0756	**0.09**	0.084	0.0979	**0.1051**
		12	0.0278	0.0734	**0.096**	0.08474	0.0935	**0.1039**
		20	0.0232	0.0966	**0.1083**	0.0856	**0.1208**	0.1055
MultiWOZ	Test	3	0.0735	0.0867	**0.0888**	**0.1016**	0.0898	0.0927
		12	0.048	**0.1148**	0.0982	0.0929	**0.1566**	0.1272
		20	0.031	**0.1687**	0.1164	0.0785	**0.2553**	0.1759

Table 9: Example outputs of Conversation History Tracker (CHT) module.

#	**Conversation History**	**Output of CHT module**	**TPP**
1	"i guess so. Do you watch espn?", "No. I don't have regular cable. I use a ROKU and I think you would have to pay extra for ESPN. I didn't watch it when I did have cable though.", "Oh, I was going to tell you about them winning an emmy but that's okay. Well, do you watch or have you watched pokemon?"	{"Pokemon": "unknown", "TV show": "unknown"}	{"Pokemon": "unknown", "TV show": "unknown"}
2	"I think I did hear something about that. I imagine it is an attempt to psych the other team out.", "So, it would be in the visiting team's locker room but not their own?", "Right. Teams do all kinds of things to bother the competition. I've heard of teams having heated benches in the winter for themselves but not for the visitors."	"sports": "no", "team activities": "no"	["sports": "unknown", "game": "unknown", "team activities": "no"]
3	"... "Yeah I heard he was criticized because of this, wonder why he did that, maybe he had it coming. ", "I don't think he saw the punch. He also won an Emmy!", "Do you happen to know why he won the emmy?"	"TV show": "yes", "winning": "yes"	"winning": "yes", "TV show": "yes", "Emmy": "yes"
4	"Maybe we will! That would help boost ratings, I would think. Maybe then they'll start letting the Saint's cheerleaders eat in the same restaurants as the players. What a stupid rule!", "That is an odd rule for sure! Kind of dumb.", "Do you a cat? They are amazing creatures. One is a mayor in Alaska!"	"pets": "yes", "cats": "yes", "Alaska cat": "yes"	"cats": "yes", "pets": "yes", "Alaska cat": "yes"

Table 10: Example outputs of Topic Recommender (TR) and Response Generator (RG) modules.

#	Output of TR module	Guideline	Output of RG module
1	"Anime","games","books"	"It is unclear if the agent_2 likes or dislikes Pokemon. agent_1's response should fall into one of the following 3 topics: ['Anime', 'games', 'books']."	"I don't watch Pokemon. I only know the basic premise and that there are over 750 episodes."
2	"sports gear", "athletic achievements", "sports injuries"	"It is unclear if the agent_2 likes or dislikes sports. agent_1's response should fall into one of the following 3 topics: ['sports gear', 'athletic achievements', 'sports injuries']."	"Yeah, I have heard that. I guess I would like to see that. Do you like any other sports?"
3	"drama", "comedy", "fantasy"	"agent_1 likes winning. agent_2's response should fall into one of the following 3 topics: ['drama', 'comedy', 'fantasy']."	"I guess he had a lot of good commercials? I love him!"
4	"exotic pets", "pets in movies", "cooking techniques"	"agent_2 likes cats. agent_1's response should fall into one of the following 3 topics: ['exotic pets', 'pets in movies', 'cooking techniques']."	"A cat named Sir Lancelot, right?"