# Chitchat as Interference: Adding User Backstories to Task-Oriented Dialogues

**Armand Stricker, Patrick Paroubek**
Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique
91400, Orsay, France
{armand.stricker, patrick.paroubek}@lisn.upsaclay.fr

**Abstract**

During task-oriented dialogues (TODs), human users naturally introduce chitchat that is beyond the immediate scope of the task, interfering with the flow of the conversation. To address this issue without the need for expensive manual data creation, we use few-shot prompting with Llama-2-70B to enhance the MultiWOZ dataset with user backstories, a typical example of chitchat interference in TODs. We assess the impact of this addition by testing two models: one trained solely on TODs and another trained on TODs with a preliminary chitchat interaction. Our analysis demonstrates that our enhanced dataset poses a challenge for these systems. Moreover, we demonstrate that our dataset can be effectively used for training purposes, enabling a system to consistently acknowledge the user's backstory while also successfully moving the task forward in the same turn, as confirmed by human evaluation. These findings highlight the benefits of generating novel chitchat-TOD scenarios to test TOD systems more thoroughly and improve their resilience to natural user interferences.

**Keywords:** Task-Oriented Dialogue, Chitchat, Automatic Data Augmentation, LLMs, Prompting, Conversational AI

## 1. Introduction

Chitchat and task-oriented dialogue (TOD) agents are typically portrayed as two distinct systems. On the one hand, chitchat agents are expected to embody all the qualities of an ideal conversationalist. They should be empathetic, engaging, knowledgeable, and well-behaved (Roller et al., 2020; Smith et al., 2020; Rashkin et al., 2019). On the other hand, task agents are designed to be efficient and effective tools (Chen et al., 2017; Deriu et al., 2021).

However, natural human communication does not make such a clear-cut distinction: most language is not purely transactional[1] or interactional[2] but rather a mix of both (Brown and Yule, 1983), and overlap is therefore common in real-world interactions.

Efforts have been undertaken to augment TOD datasets with chitchat, in order to move towards more flexible TODs (Li et al., 2023; Young et al., 2022). These augmentations are automatic as well as human-generated and produce more interesting dialogues (Stricker and Paroubek, 2023), which humans tend to prefer. However, it is important to note that each individual turn in these datasets is categorized as being strictly chitchat or task-oriented. As a result, it remains unclear how a system trained on such data might react to user turns in which chitchat and task talk are seamlessly combined (Figure 1).

These inter-mode user turns are important to consider as they can commonly occur in human-chatbot conversations. Indeed, in their analysis of live customer service chat logs, Beaver et al., 2020 noted that users often engage in self-disclosure when making requests. For example, in the travel domain, users frequently emphasize the significance of their trips by sharing details about the reasons of their travel or the people they are visiting.

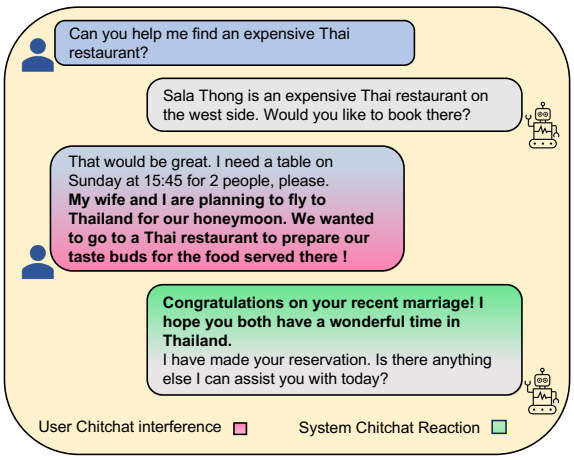


Figure 1: A chatty user incorporates elements of backstory to their task-oriented request, creating a natural interference in the TOD interaction. The system reaction accommodates the additional backstory with support and understanding, all the while avoiding the introduction of new topics. This design choice ensures that the system seamlessly transitions back to the task at hand, effectively assisting the user in achieving their goal.

[1] the focus of the encounter is external and leads to an action

[2] the focus of the encounter is internal, and is centered on the relationship between participants



*LREC-COLING 2024*, pages 3203–3214
20-25 May, 2024. 

In this work, we examine this form of interference more closely.

Although Beaver et al., 2020 have made a portion of the studied data accessible, it only includes the initial user turns from each conversation. This limitation, resulting from privacy restrictions related to live company chat logs, complicates the task of evaluating and potentially enhancing TOD agents in similar situations for researchers.

To remedy this gap, we present a dataset that includes instances where a chatty user introduces elements of backstory following their task-oriented request. In its response, the system first reacts to the user backstory before continuing with the task. Such user turns have the potential of derailing the TOD interaction, and thereby challenge the system to skillfully manage both the chitchat and the redirection of the conversation back to the task (Figure 1).

Crafting these scenarios manually is strenuous. We consequently set out to create these examples automatically, deriving them from human-generated chitchat exchanges in FusedChat (Young et al., 2022) (Section 2), a version of MultiWOZ (Budzianowski et al., 2018) augmented with full chitchat exchanges. By leveraging the information presented by the user in these exchanges, we automatically enhance a number of MultiWOZ dialogues.

To evaluate the effects of the added chitchat interferences, we utilize SimpleToD (Hosseini-Asl et al., 2020), a popular end-to-end approach that relies on a single language model (Section 3). Given the impressive performances of recent state-of-the-art models, we create a strong baseline by combining this approach with one such model, LLama-2-7B (Touvron et al., 2023), and LoRA (Hu et al., 2021), a parameter-efficient fine-tuning technique.

We use this approach to train two baseline systems. For a system trained solely on TOD, we adhere to the standard training protocol for **SimpleToD**, utilizing unaugmented MultiWOZ dialogues. To test a system that has been exposed to chitchat under a different form than in our enhancements, we train a **SimpleToD-fused** variant on FusedChat dialogues. These are MultiWOZ dialogues to which a chitchat exchange has been prepended (Appendix A). Furthermore, to estimate the effectiveness of our automatically-generated data for training purposes, we train a **SimpleToD-inter** version. We find our data can indeed allow a model to smoothly and consistently handle user backstories while advancing the task in the same turn.

Overall, we make the following contributions:

- We introduce a novel task that treats user chitchat as a disruptive element in TODs, merging task-oriented requests and elements of backstory into a single user turn.
- We automatically construct a dataset which may serve as a testbed for TOD systems.
- We carry out experiments to assess the level of difficulty posed by our task for the popular SimpleToD baseline and its variant SimpleToD-fused, trained with a state-of-the-art LLM. We outline the shortcomings and successes of each variant, drawing from both automatic and human evaluations (Sec. 4).
- We show a system trained on our automatically generated data (SimpleToD-inter) can effectively integrate chitchat reactions into its responses, moving the task forward while making the user feel heard.

Our experimental code and generated data can be found on GitHub[3].

## 2. Proposed Augmentation Pipeline

To augment the well-known MultiWOZ benchmark (Budzianowski et al., 2018), we propose to repurpose elements from the FusedChat dataset (Young et al., 2022), also a MultiWOZ extension. FusedChat both prepends and appends human-generated chitchat exchanges to the pre-existing TODs. We leverage the *prepended* exchanges to simulate our scenario, as these tend to portray a plausible context for the user's interaction, valuable for generating backstories (Appendix A).

We reduce the need for human involvement by automating the creation of backstories and chitchat reactions to add to the user and system turns, respectively. Leveraging fast and automated data generation methods has become a promising avenue in dialogue research, given the in-context learning capabilities of recent LLMs. Comparable approaches have recently been employed to create annotated TODs (Li et al., 2022), as well as chitchat dialogues grounded in personas (Lee et al., 2022). We note that compared with generating entire conversations from scratch, generating our augmentations is a more manageable process less likely to yield incoherent exchanges. Indeed, we keep the original task dialogue intact and the backstories we generate are based on contextual exchanges.

We leverage Llama-2-70B-chat[4], an open-source LLM trained on 2 trillion tokens of data from publicly accessible sources (Touvron et al., 2023). It has additionally been fine-tuned by the authors for chat. In our preliminary experiments, this particular model demonstrated higher effectiveness

[3] https://github.com/armandstrickernlp/chitchat-as-interference

[4] https://huggingface.co/meta-llama/Llama-2-70b-chat-hf

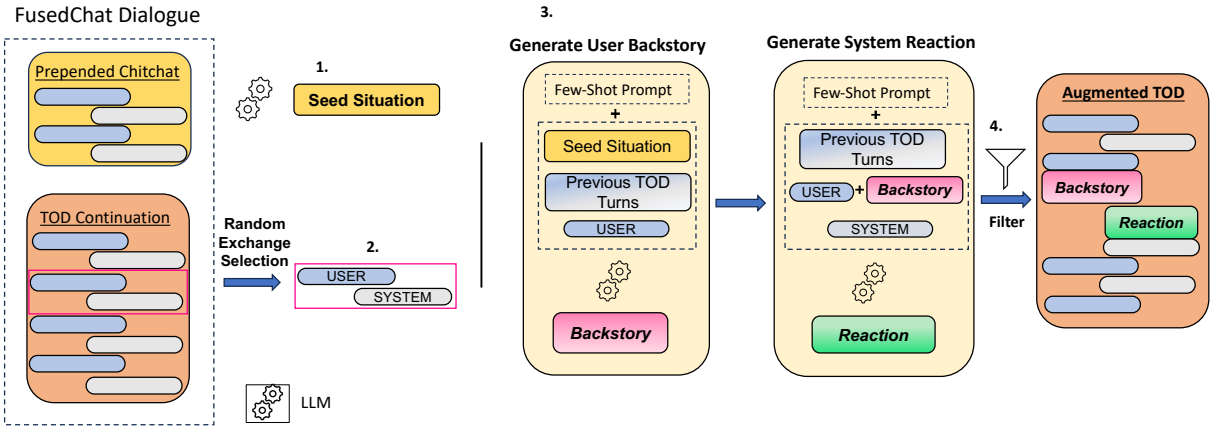


Figure 2: Our proposed augmentation generation pipeline. Given a Fusedchat dialogue, we 1) summarize the prepended chitchat into a seed situation, 2) select a random exchange to augment from the task continuation, 3) expand the exchange with in-context learning by first generating the backstory and then the chitchat reaction, 4) filter out potentially low quality generations.

in instruction-following. It also generated friendlier and more diverse system reactions compared with other Llama-2 models. Hence, we opt for this model to maximize the quality of the generated data.

We proceed in **4** main steps (Figure 2), which are applied to all examples in FusedChat that feature prepended chitchat exchanges.

**Seed Situation** We extract the prepended chitchat exchanges from FusedChat and generate summaries based on the information shared by the user. This summary serves as a seed situation that frames the interaction. We create a prompt made up of an instruction prefix and three exemplars (Figure 3).

**Turn Selection** To increase the unpredictability of the interferences, we select a random exchange from the corresponding TOD continuation. In cases where the dialogue spans multiple domains (e.g., booking a train ticket and then a hotel), we ensure that the exchange is extracted from the first sub-dialogue, which is the most related to the seed situation. We achieve this by keeping track of the dialogue acts, which carry domain information. The cutoff point is determined by the turn that introduces an act from a different domain.

**Exchange Augmentation** The structure we consider aligns with the sequences presented in Figure 1: elements of backstory are added *after* the user's original utterance and a chitchat reaction is added *before* the original system response. These components are generated in two steps. Each prompt provides an instruction and three exemplars that show the intended result. Moreover, each prompt example is structured with explicit separators to better guide the model's output.

- **Generating the user's backstory.** We pass as input to the LLM a seed situation, any preceding conversation turns, and the user's task-oriented utterance. Our prompt comprises examples that pair these inputs with a backstory which naturally follows the user's utterance (Appendix B).
- **Generating the system's reaction.** Given the previous dialogue turns, the original system response, and a newly-generated backstory, we prompt the LLM again to generate a chitchat reaction. In the prompt examples, we design the chitchat reactions to stay focused on the backstory, providing support and understanding (Appendix C). While this type of chitchat may be limited in terms of diversity, it serves the purpose of avoiding new open-domain topics which could further derail the task interaction. This, in turn, helps the system maintain efficiency in task resolution, an essential attribute for TOD systems (Deriu et al., 2021).

**Automatic filtering** Similarly to Sun et al., 2021, we additionally automatically filter out backstories and reactions that are unlikely to be of good quality. The filters reject: (i) outputs that fail to adhere to the explicit prompt structure and lack the explicit separators, (ii) outputs which contain requestable slots from MultiWOZ (phone numbers, addresses…) as these are likely to repeat information from the original task utterance (iii) outputs that exhibit similarity to the utterance being augmented or that exactly contain the utterance

| | Total unique tokens | Unique tokens not in MWOZ | Avg. num. tokens per turn |
|---|---|---|---|
| MWOZ | 18507 | — | 13.13 |
| All Augmentated Turns | 25759 | — | 28.38 |
| Turns w/ Backststory | 9419 | 6061 | 36.7 |
| Turns w/ Reaction | 4998 | 2770 | 20.06 |

Table 1: Dataset statistics

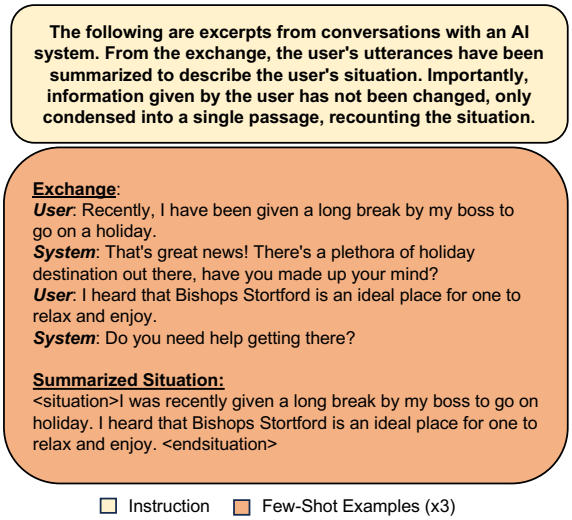


Figure 3: Few-shot prompt (3 examples) for generating the seed situation. Prompts for generating the backstory and the reaction follow a similar structure.

being augmented. We use a Levenshtein similarity ratio of 50% as a cut-off, which we found to work well in practice.

The augmented dataset contains TODs from MultiWOZ2.2 (Zang et al., 2020), augmented with our desired scenario. The dataset includes a total of **3529** training examples, **458** validation examples and **488** test examples, after filtering. Relevant statistics can be found in Table 1. The interference-augmented user turns are notably longer than the average TOD turn, reflecting the conversational style of a chatty user who shares additional details about their request. In contrast, the reactions are comparatively shorter and less diverse, as their primary purpose is to provide support and understanding rather than introduce new information.

| | Not at all | Somewhat | Fully | $\kappa$ |
|---|---|---|---|---|
| Q1 | 0% | 1.6% | 98.4% | 0.421 |
| Q2 | 0.8% | 12.3% | 86.9% | 0.236 |
| Q3 | 6.6% | 12.3% | 81.1% | 0.341 |

Table 2: Averages for each rating on each question in %. We also calculate Fleiss's $\kappa$ for each question to estimate inter-rater agreement.

**Human Evaluation** To control the quality of our examples, we perform an in-house human annotation of 25% of the test set (a sample of 122 exchanges picked at random). We bring on 3 annotators who all have a background in linguistics, enabling them to potentially detect subtle inconsistencies in the exchanges. For each example, participants are presented with the generated user situation, the previous turns leading up to the augmented exchange and the exchange to be rated. Raters assess the exchange along 3 dimensions:

- *Q1. In the user turn, is the backstory being presented reasonable given the situation?* This question assesses whether the model has followed instructions and based the user's backstory on the situation. If the model generates content not explicitly present in the seed situation, raters are instructed not to penalize the output as long as it remains reasonable and coherent. For example, if the LLM adds *I'm excited to go back!* to a backstory where the user is returning home to visit their family, the output should receive full marks.
- *Q2. In the system's response, is the reaction provided supportive and understanding of the user's backstory ?* This dimension evaluates whether the LLM adheres to instructions and generates an appropriate reaction, consistent with our design choice.
- *Q3. Overall, does the exchange sound natural and coherent ?* Finally we ask annotators to rate the exchange as a whole, to ensure it retains its original coherence post-augmentation.

For each dimension, raters must choose between the following labels: *Not at all, Somewhat,* and *Fully*.

Examining Table 2, human annotations reveal that the LLM effectively follows the given prompts and creates coherent exchanges overall, with fair to moderate agreement between annotators (Fleiss's $\kappa$). An average of only 6.6% examples are rated as incoherent. Upon inspection of the selected exchanges, this rating arises when (i) the backstory amplifies an incoherence present in the original exchange (e.g. the original task response provides incorrect train times while the backstory emphasizes the importance of punctuality), (ii) when the

randomly selected exchange makes it challenging to seamlessly incorporate the backstory (e.g. the backstory involves booking a museum in London while the selected exchange focuses on finding a museum in Cambridge) and (iii) the system reaction lacks commonsense (e.g. the reaction suggests that the airport is the user's final holiday destination). We treat these examples as noise to be expected in the dataset.

Overall, this readily implementable approach yields acceptable and natural exchanges as rated by humans, and lays the groundwork for future augmentation endeavors with LLMs for combining chitchat and TOD.

## 3. Methodology and Experimental Setup

### 3.1. Methodology

We choose the SimpleToD method (Hosseini-Asl et al., 2020) to implement our baselines. Although the approach is typically trained with GPT-2 (Radford et al., 2019), we replace it with the more recent and state-of-the-art Llama-2-7B[5] (Touvron et al., 2023), supplemented with LoRA (Hu et al., 2021). This parameter-efficient technique has been shown to produce results comparable to full fine-tuning (Hu et al., 2021) and is therefore state-of-the-art for fine-tuning LLMs. It also produces a more regularized model, since a large portion of the pre-trained weights are left intact. This helps when facing unseen challenges during inference and makes this a powerful and effective method. We note that, as of yet, in-context learning approaches have not proven as effective for end-to-end TOD (Hudeček and Dusek, 2023; Si et al., 2024). We therefore adhere to the practice of fine-tuning in this work.

The SimpleToD method is a recent and popular end-to-end approach which relies on a single language model. We choose this method as it has been employed in prior research that explores chitchat-augmented TODs (Chen et al., 2022; Sun et al., 2021) and because it follows a very similar procedure to other strong baselines (Ham et al., 2020; Yang et al., 2021).

The SimpleToD method only takes as input the previous dialogue turns to sequentially generate three task-oriented sub-components in the form of text (Figure 4).

- First, the user's constraints are generated in the form of [domain, slot, slot_value] triplets. One possible example is [restaurant, food, indian].

[5] https://huggingface.co/meta-llama/Llama-2-7b

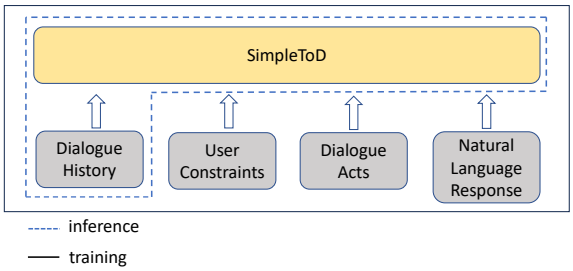


Figure 4: An overview of the SimpleToD approach. At training time, a sequence of components is fed into a generative language model. During inference, only the dialogue history is used as input, and each component is generated in step-by-step autoregressive fashion.

- Subsequently, a collection of dialogue acts is produced, in the form of [domain, act, slot]. One possible example is [restaurant, inform, name].

- Finally, the model generates a response for the user (*The name of the restaurant is The Golden Curry*). To ensure system adaptability to various databases, the model is often trained on *delexicalized* responses, where entities to be fetched from the database are replaced by placeholders (*The name of the restaurant is [name]*). This also simplifies the evaluation process when verifying the type of entity being offered by the system.

The language model is fine-tuned on instances where [*Dialogue History, User Constraints, Dialogue Acts, Response*] are concatenated into a single text sequence using special tokens as separators. During inference, we only pass [*Dialogue History*] as input and the model is expected to generate the rest of the sequence.

### 3.2. Experimental Setup

We propose two baseline variants and a model trained on our own augmented data, using the SimpleToD approach. The conditions for training and testing each variant are detailed in Table 3. Distinctions between models are based on training data augmentations, as identical dialogue IDs are used in each set..

- **SimpleToD** is a baseline representative of a typical task-oriented system: it is trained as intended by the SimpleToD authors. For reference, **SimpleToD-ref** is trained in the same way, but *tested* on MultiWOZ examples.

- **SimpleToD-fused** is a baseline representative of a more versatile model, exposed to both TOD and chitchat turns during training. This model is trained on FusedChat examples

which include prepended chitchat exchanges. Following the FusedChat authors' two-in-one approach, this model is trained to generate a task response when provided with a task input, and a chitchat response when given a chitchat input..

- **SimpleToD-inter** is trained on our own augmented examples. We use this model to verify that a model trained on our synthetic data can indeed 1) learn the MultiWOZ tasks adequately and 2) accommodate user chitchat with support and understanding while advancing the task in the same turn. No dataset has enabled models to do so thus far.

| Model | Train | Test |
|---|---|---|
| SimpleToD-ref | MWoz | MWoz |
| SimpleToD | MWoz | Interfere |
| SimpleToD-fused | FChat | Interfere |
| SimpleToD-inter | Interfere | Interfere |

Table 3: The same core MultiWOZ dialogues are used to train all models. *MWoz* denotes the original, unaltered MultiWOZ dialogues; *FChat* signifies dialogues are augmented as in FusedChat; and *Interfere* indicates dialogues are enhanced with our custom augmentation. Distinctions between models are based on training data augmentations. Identical dialogue IDs are used across comparisons.

**Implementation** We train all models with the huggingface[6] and pytorch (Paszke et al., 2017) frameworks. For each variant, we perform a grid search over learning rates ranging from 1e-5 to 1e-4 with an effective batch size of 32 (batches of 16 with 2 gradient accumulation steps). We use learning rates of 3e-5 for all variants and train for 4 epochs. When performing LoRA fine-tuning, we choose the following LoRA configuration: a rank of size 32, given the complexity of our task, a scaling factor of 32, and target modules which include query and value projection matrices, found in the self-attention module of each transformer block. Training and inference is done on a single 80Gb A100 GPU using 4 random seeds for each experiment. After training, we generate outputs using greedy decoding as in SimpleToD and prevent the model from repeating n-grams of size 10, to avoid repetitive loops that may arise.

**Automatic Evaluation Metrics** We employ the standard evaluation metrics used for MultiWOZ[7] (Nekvinda and Dušek, 2021). The **joint goal accuracy** (JGA) reflects the proportion of turns where the predicted user constraints exactly match the gold ones. The **inform rate** evaluates the system's capacity to provide the right type of entities from the database, given the user's constraints, and the **success rate** assesses how effectively the system delivers requested attributes like phone numbers or booking references. For a more comprehensive understanding of these metrics, we direct readers to the MultiWOZ paper (Budzianowski et al., 2018). For response quality, we use three BLEU (Papineni et al., 2002) scores as in (Chen et al., 2022): **BLEU-aug** rates responses that follow interferences, **BLEU-orig** evaluates all other responses and **BLEU-all** evaluates all responses holistically. For response diversity, we analyze the **Conditional Bigram Entropy** (CBE) and the count of **unique trigrams** to gauge the richness of vocabulary and phrasing in the generated responses (Nekvinda and Dušek, 2021).

## 4. Results and Discussion

**Automatic Evaluation Results** The results of the automatic evaluation in Table 4 suggest that our chosen **SimpleTOD baseline** is quite robust to user backstories. Despite the interferences injected in the test set, SimpleToD perfoms as well as SimpleToD-ref, staying on track with regards to the task. We expect this baseline's robustness to be related to our choices of model and fine-tuning technique. Llama-2 and LoRA's regularization-friendly setup aids in maintaining task focus when confronted with novel scenarios like the interferences we introduce. However, BLEU-aug shows that responses to user turns with chitchat interferences are not satisfactory, as we further show in our human evaluation below.

On the other hand, **SimpleToD-fused** encounters challenges when presented with the interference-augmented test set. This model is specifically trained to respond to either a chitchat turn or a task-oriented turn, and as a result, falls into one mode or the other when confronted with our interferences. In most cases, it leans towards continuing with chitchat, resulting in lower inform and success rates. This also results in very low BLEU-aug scores, primarily because the TOD segment is frequently absent from the response, and also because the chitchat that is generated does not closely match the expected reactions. This reveals that training on dialogues where chitchat and TOD are depicted as mututally exclusive response modes falls short in addressing inter-mode user turns.

The **SimpleToD-inter** model, trained on our augmented data, delivers task-oriented performance

[6] https://huggingface.co

[7] https://github.com/Tomiinek/MultiWOZ_Evaluation?tab=readme-ov-file

| | inform | success | JGA | CBE | unique tris. | BLEU-aug | BLEU-orig | BLEU-all |
|---|---|---|---|---|---|---|---|---|
| SimpleToD-ref | 80.73 | 72.85 | 0.66 | 1.87 | 2870 | – | – | – |
| SimpleToD | 80.47 | 72.85 | **0.66** | 1.91 | 3184 | 0.11 | **0.24**$^*$ | 0.21 |
| SimpleToD-fused | 67.08 | 51.38 | 0.39 | 2.12 | 5674 | 0.04 | 0.2 | 0.17 |
| SimpleToD-inter | **82.07** | **73.2** | 0.64 | **2.21** | **7561**$^*$ | **0.25**$^*$ | 0.22 | **0.22** |

Table 4: Automatic evaluation results (baselines described in Sec. 3). Values in bold indicate the best value for each metric and * indicates a statistically significant difference ($p < 0.05$, paired t-test) between the best and the second best values in each respective column.

| | #1 | #2 | #3 | Mean Rank |
|---|---|---|---|---|
| SimpleToD | 11.33% | **77.33**% | 11.33% | 2.0 |
| | ± 10.37 | ± 7.36 | ± 4.99 | ± 0.15 |
| SimpleToD-fused | 5.33% | 33.33% | **61.33**% | 2.56 |
| | ± 2.49 | ± 10.5 | ± 10.87 | ± 0.12 |
| SimpleToD-inter | **92.0**% | 6.67% | 1.33% | **1.09** |
| | ± 3.27 | ± 3.4 | ± 0.94 | ± 0.03 |

Table 5: Human rankings of responses from each baseline, following a user request augmented with backstory. The percentages represent the distribution of rankings per model. Rank 1 is best, so a lower rank implies preferred responses.

comparable to that of SimpleToD, with no statistically significant differences observed. Moreover, it aligns as expected with the augmented system responses, as indicated by the Bleu-aug score. Consequently, this shows that our generated training data enhances the model's adaptability to potentially chatty users, without negatively impacting its task performance.

**Human Evaluation Results** To better evaluate the output quality of our variants, we conduct an in-house human evaluation of delexicalized responses to user turns with interferences. Three evaluators, distinct from those used in our initial annotation, participate in this task. They all possess expertise in NLP, making delexicalized responses pose no issue. To create a simple and straightforward annotation, we adopt a methodology inspired by Nekvinda and Dušek, 2022 which consists in side-by-side relative ranking. This approach has demonstrated enhanced consistency when contrasted with evaluating individual instances in isolation (Kiritchenko and Mohammad, 2017). We randomly select 50 examples from the test set, singling out the turns that contain our introduced interferences.

For each example, participants are presented with the dialogue history up to the augmented user turn and three system responses to be ranked. The responses are displayed side by side, each of them having a dedicated scale from 1 to 3, indicating the quality from best to worst. Annotators are explicitly instructed to focus on how well each response helps the user with their booking *and* on how naturally the response follows the previous user turn. If the evaluators deem it relevant, they can assign the same rating to multiple responses.

Results in Table 5 show a persistent preference for SimpleToD-inter: this model's responses both move the task forward and follow more naturally after the user's turn. Regarding annotator agreement, we find that standard deviations offer more insight than Fleiss's $\kappa$, and note that there is high agreement on ranking SimpleToD-inter responses as the top choice (#1). However, the level of agreement is lower for the other two baselines, even though there is a clear inclination toward the responses from SimpleToD.

We inspect the rated responses, showing some illustrative examples in Table 6. To characterize each variant, we employ Taleb, 2014's Fragile-Robust-Antifragile framework, which has previously been applied in the context of Information Systems (Gorgeon, 2015). We adapt this framework to assess how each variant handles inter-mode inputs that potentially disrupt the flow of a TOD.

Responses from SimpleToD-fused typically belong to the "Fragile" category. This model, when presented with a user backstory after a request, often produces responses focused more on chitchat, neglecting the original request. Such behavior slows down the task-oriented dialogue and increases the likelihood of the conversation deviating from its intended path.

SimpleToD can be classified as relatively "Robust." It predominantly disregards the elements of backstory in the user turn and maintains a strong focus on accomplishing the task. However, this approach results in responses that may appear detached and dismissive of the user, leaving a negative impression that could dissuade users from returning to the system. Interestingly, in some rare instances, the model demonstrates the capacity to generate supportive reactions such as *I'm sorry to hear about...* before reorienting its answer back to the task, as seen in example #3, Table 6. This observation can be attributed to our model and fine-tuning choices.

To an extent, SimpleToD-inter demonstrates char-

| *Dialogue History* | *Responses* |
|---|---|
| *User*: I need a taxi. I am going to Avalon and I need to leave after 16:15<br>*System*: I will need to know where you need to be picked up?<br>*User*: I will be at Parkside Pools. I'm so excited for later! My crush just invited me to her birthday party at her house, Avalon. I'm super nervous right now! | **SimpleToD**: I have booked a [type] for you. The contact number is [phone].<br>**SimpleToD-fused**: I hope you have a great time!<br>**SimpleToD-inter**: Aww, that's so sweet! I'm sure you'll have a great time at the party. I have a [type] for you. The contact number is [phone]. |
| User: I'd like to take a train from Cambridge on Sunday...<br>System: ...TR7447 will get you there by 13:51. Do you need some seats booked?<br>User: Yes. Can you please book 2 seats? I'm actually going to the dentist on Sunday. I have a toothache and I'm hoping to get it fixed. I'm trying to soothe the pain with ice until then... | **SimpleToD**: Booking was successful, the total fee is [price] payable at the station. Reference number is: [ref]. Is there anything else I can help you with?<br>**SimpleToD-fused**: Booking was successful, the total fee is [price] payable at the station. Reference number is: [ref]. Is there anything else I can help you with?<br>**SimpleToD-inter**: Sorry to hear that you're in pain. I hope the dentist appointment goes well and that you get some relief from the toothache. I have booked [bookpeople] tickets for you. Your reference number is [ref] . Is there anything else I can help you with? |
| User: I am departing from Cambridge on Friday. I'm actually going to visit my friend in the hospital. He got into a motorbike accident and I've been worried sick about him. I've always told him that motorbiking is dangerous, but he wouldn't listen. Now he's paying the price. I hope he's doing alright... | **SimpleToD**: I'm sorry to hear about your friend. I'm sure he'll be ok. I have train [trainid] arriving at [arriveby] . Would you like me to book it for you?<br>**SimpleToD-fused**: I'm sorry to hear about your friend. I hope he's okay.<br>**SimpleToD-inter**: Sorry to hear about your friend. I hope he recovers quickly. [trainid] leaves at [leaveat] and arrives at [arriveby]. Would you like to book a seat? |

Table 6: Examples of system responses following augmented user turns (dialogues SNG0055, PMUL0994 and PMUL1424). These serve to demonstrate how SimpleToD-fused alternates between task-oriented and chitchat responses, with chitchat being predominant. Example #3 highlights SimpleToD's occasional capacity to offer support alongside addressing the task request. However, this behavior is not as consistent as in the case of SimpleToD-inter, the model favored by human raters and trained on our augmented dataset.

acteristics that align with being "Antifragile," a term which refers to being able to benefit from potential disturbances. In this context, inter-mode inputs are welcomed and handled accordingly, enhancing the overall interaction compared to more typical TOD flows. This creates a more engaging dialogue and ultimately incentivizes the user to continue using the system.

Nonetheless, in terms of limitations, we note that responses exhibit somewhat limited diversity, which could lead to users anticipating the system's reactions. Moreover, while we have addressed one specific scenario, the challenge of making a single system antifragile to a variety of scenarios remains. However, given our encouraging results, generating inter-mode scenarios automatically and then training a TOD system in a multi-task fashion holds promise.

## 5. Related Work

The integration of chitchat into task-oriented dialogues is a growing trend in recent research. For instance, Accentor (Sun et al., 2021) adds automatically generated, human-filtered chitchat to system responses. However, their chitchat often lacks specific contextual relevance to the ongoing conversation, resulting in responses like *Sounds good!*. On the other hand, KETOD (Chen et al., 2022) expands system responses by introducing knowledge-grounded chitchat. This chitchat relies on specific, contextually relevant information, leading to more diverse enhancements compared to Accentor. While the knowledge is automatically extracted from Wikipedia, system responses are refined by human annotators. Although valuable, both of these approaches have limitations as they exclusively augment system turns, assuming the user input will stay fixed.

Other work takes an interest in adding full chitchat exchanges to TODs. For example, Fused-Chat (Young et al., 2022) prepends and appends human-generated chitchat. Recently, Liu et al. build on this dataset to explore system-initiated transitions from chitchat to task-oriented dialogue. Additionally, Li et al. automatically inject chitchat exchanges within MultiWOZ dialogues, creating more engaging and flexible TODs. While these contributions are valuable, they typically treat individual turns as exclusively task-oriented or chitchat-oriented, without allowing for potential overlap. Our work represents a step towards addressing more ambiguous user turns that may contain elements of both modes simultaneously.

## 6. Conclusion

We portray chitchat in TODs as a natural interference, introducing a novel task in which users incorporate elements of backstory into their requests. We generate this scenario in an automated and controlled manner by employing few-shot prompting with a large language model. This method produces exchanges human evaluators deem acceptable and natural. Our study reveals that a commonly used TOD baseline cannot fully handle this setting and neither can its variant trained on a mixture of TOD and chitchat turns. In contrast, a model trained on our enriched dataset consistently and seamlessly handles the introduced chitchat while effectively advancing the task at hand. We believe our work is an important step in automating the creation of diverse and challenging chitchat scenarios within TODs, ultimately leading to the emergence of more resilient TOD systems.

## Ethics Statement

Our dataset is the product of automated generation, which means it could potentially contain biases from the data used to train the underlying LLM. Nevertheless, this LLM is open-source, making it possible for the wider research community to thoroughly examine these biases. Furthermore, it is important to note that our training data is not guaranteed to exclusively contain supportive responses. Therefore, it should be employed judiciously, being better suited for low-stakes settings or research purposes.

## 7. Bibliographical References


Ian Beaver, Cynthia Freeman, and Abdullah Mueen. 2020. Towards Awareness of Human Relational Strategies in Virtual Agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2602–2610.

Gillian Brown and George Yule. 1983. *Teaching the spoken language*, volume 2. Cambridge university press.

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.

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

Zhiyu Chen, Bing Liu, Seungwhan Moon, Chinnadhurai Sankar, Paul Crook, and William Yang Wang. 2022. KETOD: Knowledge-enriched task-oriented dialogue. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2581–2593, Seattle, United States. Association for Computational Linguistics.

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54:755–810.

Arnaud Gorgeon. 2015. Anti-fragile information systems. In *Proceedings of the International Conference on Information Systems - Exploring the Information Frontier, ICIS 2015, Fort Worth, Texas, USA, December 13-16, 2015*. Association for Information Systems.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A Simple Language Model for Task-Oriented Dialogue. In *Advances in Neural Information Processing Systems*, volume 33, pages 20179–20191. Curran Associates, Inc.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Vojtěch Hudeček and Ondrej Dusek. 2023. Are large language models all you need for task-oriented dialogue? In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–228, Prague, Czechia. Association for Computational Linguistics.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*

*(Volume 2: Short Papers)*, pages 465–470, Vancouver, Canada. Association for Computational Linguistics.

Young-Jun Lee, Chae-Gyun Lim, Yunsu Choi, Ji-Hui Lm, and Ho-Jin Choi. 2022. PERSONACHATGEN: Generating personalized dialogues using GPT-3. In *Proceedings of the 1st Workshop on Customized Chat Grounding Persona and Knowledge*, pages 29–48, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Miaoran Li, Baolin Peng, Michel Galley, Jianfeng Gao, and Zhu (Drew) Zhang. 2023. Enhancing task bot engagement with synthesized open-domain dialog. In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*, pages 496–508, Prague, Czechia. Association for Computational Linguistics.

Zekun Li, Wenhu Chen, Shiyang Li, Hong Wang, Jing Qian, and Xifeng Yan. 2022. Controllable dialogue simulation with in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4330–4347, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ye Liu, Stefan Ultes, Wolfgang Minker, and Wolfgang Maier. 2023. 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.

Tomáš Nekvinda and Ondřej Dušek. 2021. Shades of BLEU, flavours of success: The case of MultiWOZ. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 34–46, Online. Association for Computational Linguistics.

Tomáš Nekvinda and Ondřej Dušek. 2022. AARGH! end-to-end retrieval-generation for task-oriented dialog. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 283–297, Edinburgh, UK. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Stephen Roller, Y-Lan Boureau, Jason Weston, Antoine Bordes, Emily Dinan, Angela Fan, David Gunning, Da Ju, Margaret Li, Spencer Poff, Pratik Ringshia, Kurt Shuster, Eric Michael Smith, Arthur Szlam, Jack Urbanek, and Mary Williamson. 2020. Open-domain conversational agents: Current progress, open problems, and future directions.

Shuzheng Si, Wentao Ma, Haoyu Gao, Yuchuan Wu, Ting-En Lin, Yinpei Dai, Hangyu Li, Rui Yan, Fei Huang, and Yongbin Li. 2024. Spokenwoz: A large-scale speech-text benchmark for spoken task-oriented dialogue agents. *Advances in Neural Information Processing Systems*, 36.

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030, Online. Association for Computational Linguistics.

Armand Stricker and Patrick Paroubek. 2023. Enhancing task-oriented dialogues with chitchat: A comparative study based on lexical diversity and divergence. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Kai Sun, Seungwhan Moon, Paul Crook, Stephen Roller, Becka Silvert, Bing Liu, Zhiguang Wang, Honglei Liu, Eunjoon Cho, and Claire Cardie. 2021. Adding chit-chat to enhance task-oriented dialogues. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1570–1583, Online. Association for Computational Linguistics.

Nassim Nicholas Taleb. 2014. *Antifragile: Things that gain from disorder*, volume 3. Random House Trade Paperbacks.

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*.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14230–14238.

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.

## A. Prepended Chitchat in FusedChat

Refer to Table 7.

| | |
|---|---|
| User (*CC*) | I am meeting my client in Cambridge soon. I'm kind of nervous. |
| System (*CC*) | Is it an important meeting? |
| User (*CC*) | This is my first client. |
| System (*CC*) | Wow that is huge! Good luck! |
| User (*Task*) | I am looking for a train that will be arriving there by 16:00 Friday, from King's Lynn. |
| System (*Task*) | We have a train headed for Cambridge at 15:11. Would you like to book it? |
| ... | ... |

Table 7: A full chitchat exchange, related to the task, is prepended by human annotators to a MultiWOZ dialogue. *CC* indicates chitchat turns and *Task* indicates task-oriented turns. The ellipses indicate the continuation of the original MultiWOZ dialogue.

## B. User Backstory Prompt

Refer to Table 8.

| |
|---|
| In the following examples, you are presented with a user's situation and a conversational context, which may be None if it is the start of the conversation. The user shares their backstory by adding it to their original utterance. Their backstory is based on the user's situation and should naturally follow the original utterance. It should be very fluent and coherent with the conversational context. |
| Situation:<br>I was talking to my brother on the phone earlier today. He's getting married We discussed his wedding plans and decided to meet up at the London Liverpool Street train station today. |
| Conversational Context:<br>**User**: I would like for a taxi to take me to london liverpool street train station, arriving no later than 17:45 please.<br>**System**: I can book that for you, first I'll need to know where you'd like picked up at. |
| Original User Utterance:<br>**User**: **I would like to depart from London Kings Cross Train Station.** |
| User Utterance With Backstory:<br>**User**: **I would like to depart from London Kings Cross Train Station.** + <Backstory: My brother is getting married! I was talking to him on the phone earlier and we decided to meet at the London Liverpool train station.> [END] |

Table 8: An example prompt for generating backstories. The first row contains the instruction prefix, added at the start of the prompt. All subsequent rows represent the components of the prompt, added in that order, to create a single exemplar. Our prompt contains 3 such exemplars.

## C. System Reaction Prompt

Refer to Table 9.

| |
|---|
| In the following examples, you are presented with a conversational context. In the last turn, the user shares their backstory. The original system response should be improved to include a reaction to the user's backstory at the beginning of the response. This reaction should be supportive and display an understanding of the user's situation. It should be unique to the backstory and contextual to the conversational context. Avoid repeating expressions found in previous examples. |
| Conversational Context:<br>**User**: I would like for a taxi to take me to london liverpool street train station, arriving no later than 17:45 please<br>**System**: I can book that for you, first I'll need to know where you'd like picked up at.<br>**User**: I would like to depart from London Kings Cross Train Station. <Backstory: My brother is getting married! I was talking to him on the phone earlier and we decided to meet at the London Liverpool train station.> |
| Original System Response:<br>**System**: **A white Toyota is booked for you.** |
| Response With Reaction:<br>**System**: <Reaction: I see! Congratulations to your brother!> + **A white Toyota is booked for you.** [END] |

Table 9: An example prompt for generating chitchat reactions. The first row contains the instruction prefix, added at the start of the prompt. All subsequent rows represent the components of the prompt, added in that order, to create a single exemplar. Our prompt contains 3 such exemplars.