FRAPI: A Framework for Generating Competitive Test Beds for Task-Oriented Dialog Systems

Anonymous ACL submission

Abstract

Current test sets for task-oriented dialog systems tend to overestimate the systems' performance on conversation-level tasks like dialog state tracking. We observed that they fail to showcase similar efficacy when tested on some commonly occurring realistic scenarios like repetition and clarification through dialogues. This limited generalizability of models can be attributed to two key aspects. Firstly, crowd-workers who create these test sets have a highly restrictive/limited dialog policy to generate samples, leading to very rigid and less realistic samples. Secondly, the train and test splits are plagued with annotator biases since the same set of crowd-workers is recruited to create both splits. Using a graphical framework for dialogues, called Conversation Flow Modeling, we highlight the limitations for one such dataset. While motivating practitioners to create stricter test sets, we propose FRAMEWORK FOR AUTOMATED PATTERN INDUCTION (FRAPI), an HCI (human-computer-interaction) framework for the induction of additional natural dialog flows. FRAPI helps create annotator-bias-free patterns in testbeds of task-oriented dialog systems with minimal human intervention. Using FRAPI, we build a testbed for the models trained on the MultiWOZ data set. The proposed testbed helps validate learning from diverse yet natural patterns. Through it, we highlight the shortcomings of the current architectures to model simple, realistic human-level language variations on dialog state tracking.

1 Introduction

When task-oriented dialogue systems are evaluated using standard metrics on common data sets like the MultiWOZ dataset (Budzianowski et al., 2018), there is usually an overestimation of performance on the provided test bed. This can be attributed to the model *learning the data, and not the task* (Linzen, 2020). In an ideal world, NLP data would be a good representative of the task. However, in reality, these data sets are plagued with local biases. In the machine learning community, this problem is usually referred to as the *Generalization* problem. The biases can range from data sets adhering to very limited patterns to data sets having certain annotator biases (Geva et al., 2019). To alleviate such biases and integrate more of the natural language nuances into the testing protocol, Ribeiro et al. (2020) propose CHECKLIST a software-engineering motivated approach to evaluate the current NLP systems. This approach not only helps in assessing models, but also in developing competitive test beds.

While CHECKLIST is helpful in providing sentence level modifications for *turn-level* tasks like Intent classification (IC) and Slot Labeling (SL) (Liu and Lane, 2016; Goo et al., 2018; Wang et al., 2018b; Gupta et al., 2019), it provides no clear guidelines on how to approach the *conversational-level* problems i.e., tasks that leverage contextual information e.g., Dialog Generation and Dialog-State-Tracking (DST). (Williams et al., 2005; Williams and Young, 2007; Wu et al., 2019a). Hence, the applicability of this framework is limited when it comes to task-oriented dialogue systems. We propose a complementary approach FRAMEWORK FOR AUTOMATED PATTERN INDUCTION (FRAPI) which helps in constructing competitive test beds specifically dealing with *Conversational-level* problems. Our contributions are as follows:

1. Introduce a mechanism for analyzing failure patterns in a data set via conversation flow modeling. We do this for MultiWOZ2.1 (Eric et al., 2019).

2. Leverage a two-person spoken conversation data set: TaskMaster (Byrne et al., 2019) to infuse natural but more *complex patterns* into conversational data sets.

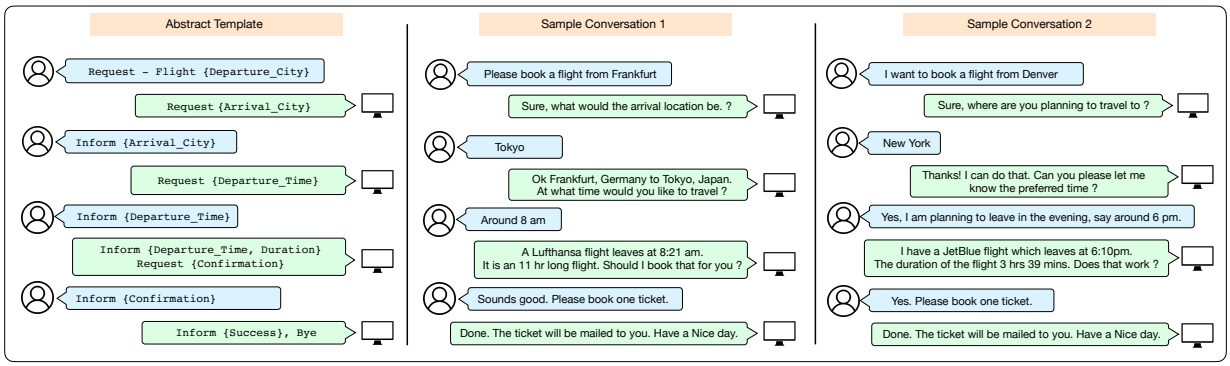

Figure 1: Both Conversation 1 and 2 are natural language realizations of the same abstract conversation flow template given on the left. They differ only in their surface forms but each turn essentially contains/asks for the same information. Please refer Section 3 for details

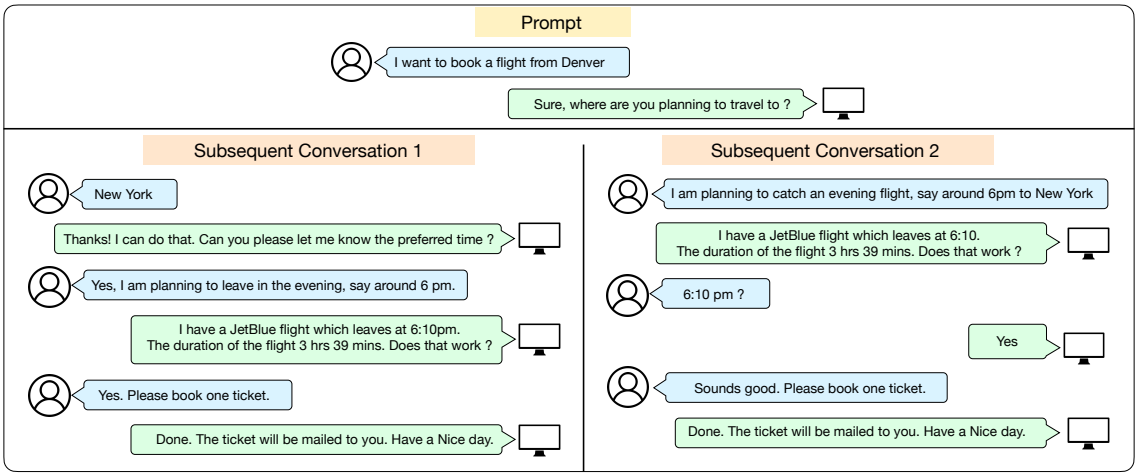

Figure 2: Although the goal of both the conversation is the same - Flight booking. They differ in the overall conversation flow abstraction to achieve the same goal. The first two turns (called prompt) are the same for both the conversations and based on the human unpredictability, follow different *conversation flow paths* in each conversation. Please refer Section 3 for details

3. Propose FRAPI, a framework for building competitive test-points for task-oriented dialog systems with minimal human supervision.

4. Demonstrate the applicability of the proposed framework on the MultiWOZ2.1 test set. We build a competitive test bed MTASK-TEST, incorporating those realistic *complex patterns*.

2 Related Work

Conversational Flow modeling via graph data-structures has previously been explored in Gritta et al. (2020). We adopt a similar approach albeit with the difference in the application. While Gritta et al. (2020) uses graph sampling for augmentation to help with dialog-policy/policy learning, the focus of our work is to produce diversity-rich and competitive natural language samples with annotations. We anticipate a lot of future research scope using conversational flow graphs.

Dialogue State Tracking (DST) is an important task in goal-oriented dialogue systems. Correct dialogue state tracking helps the agent construct a coherent response and helps, in-directly, resolve long-term dependency issues in conversations, where the *long term* is across all the previous dialogue turns. Other pertinent issues like *believability* of a conversation i.e., how realistic is the complete conversation, also depend directly on the dialogue-state. An example of unrealistic/unbelievable conversation might look like, *"I am looking for a Lufthansa flight to the Moon"*.

There have been multiple previous works (Wu et al., 2019b; Zhou and Small, 2019; Heck et al., 2020) which focus on the problem of dialogue-state tracking. Our work focuses on highlighting the shortcomings of two of the current models (Zhou and Small, 2019; Heck et al., 2020) through natural

policy variations.

Generalization through Data. Another line of work which deals with the notion of generalizability is DialoGLUE (Mehri et al., 2020). DialoGLUE groups together different types of tasks in dialogue systems and is focused on testing the generalizability of a unified model, as in GLUE (Wang et al., 2018a). In contrast, we aim to provide a framework for assessing conversation-flow robustness of models for a single conversational-level task at a time, be it dialogue generation or dialogue state tracking.

Shah et al. (2018) propose a bootstrapping mechanism for generating data sets having high coverage w.r.t dialog/conversations flows. Their main goal is to generate turn-level template guidance for crowd-source workers. These templates, however, are not natural language texts. Hence each template is sent to crowd-workers for conversion to natural language. In contrast, our proposed framework provides natural language utterances, thereby reducing the burden on crowd-workers. With our approach, the task of crowd-workers essentially reduces to performing minor edits in the dialogue state annotations.

Realistic variations and Standard Test-bed. Diverse decoding methods (Vijayakumar et al., 2018; Kumar et al., 2019), as well as CHECKLIST (Ribeiro et al., 2020) might offer linguistic variability at turn level however they do not provide a way to do it at the conversational level. Our goal is not to compete with their work, but to offer a complementary approach which is suited for creating competitive test sets task-oriented dialogue systems.

Ganhotra et al. (2020) highlight the effects of inducing naturalistic patterns in Goal-Oriented Dialog. While, the naturalistic patterns help in assessing the robustness of the systems, the approach provided in the paper is mostly restricted to simpler datasets like the bAbI dataset (Bordes et al., 2016), not easily scalable to new domains and locales, and requires a lot of manual effort to incorporate (and potentially annotate) the pattern into a conversation. In contrast, our framework provides a scalable approach while looking at a more complex dataset i.e., MultiWOZ2.1.

In the next section, we understand conversation flow and some problems associated with it.

3 Understanding Conversation Flow

Every conversation (be it human-human or machine human), despite being fraught with uncertainties (in terms of human unpredictability or machine understanding) as well as linguistic variability, has a certain level of underlying abstract **Conversation Flow**. At a high level, a **Conversation Flow** governs *how a conversation proceeds* and is not concerned with the linguistic variability associated with each turn in the conversation. We elucidate the importance of conversation flow using two main examples:

1. **Linguistic Variability**: Consider the two parallel samples (Sample 1, Sample 2) in Figure 1. We can see that the two conversations, though addressing the same problem and associated with similar domains, differ a lot in their surface forms. However on an abstract level, the conversation flow of both these conversations is very similar, as depicted on the left in Figure 1.

2. **Uncertainty**: Consider the two conversations in Figure 2. We can see that though the two conversations have the same end goal, the way the conversation proceeds is very *different*. In sample (a), the user answers every question asked by the agent, perfectly; no more no less. However, in sample (b) the user, asks for clarification, and provides multiple slots even without the specific agent prompt. These conversations, though realistic, showcase the unpredictability of humans when it comes to providing relevant information.

Dialog Policy Learning v/s Conversation Flow

Dialog Policy Learning is the task of assigning a probability to possible dialog acts based on the conversation history. The realm of policy learning is restricted to agent actions only. In contrast, conversation flow modeling is concerned not only with the possible choices for an agent but also with the unpredictability of humans. In essence, conversational flow modeling is a more complicated task and one which encompasses dialog policy.

The primary goal of this work is to target the **uncertainty** associated with the conversation flow, and build a competitive test-set which has instances of those uncertainties. To analyze conversation flows, we use an abstract graph data-structure which we describe in the next section. Insights from this analysis is essential for building FRAPI.

4 Conversation Flow Modeling

In this section, we describe the abstract framework used for analyzing the conversation flows. Using the abstraction, namely the conversational flow graph, we partition the MultiWOZ2.1 *test set* into two disjoint sets. We then analyze the sets individually to find out key characteristics of each set. The analysis plays a pivotal role in building the competitive test set.

4.1 Dataset

We use the MultiWOZ2.1 dataset (Eric et al., 2019), a multi-domain dialogue dataset spanning 7 distinct domains like `attraction`, `hotel`, `restaurant`, `taxi`, `train`, for obtaining and analysing conversational flow. It is a consolidated dataset build on top of MultiWOZ2.0 (Budzianowski et al., 2018) with relevant state corrections and added dialogue act annotations. It contains a collection of fully-labeled (intent-slot labeling and dialogue state tracking) human-human *written* conversations gathered using the WOZ (Wizard-of-Oz) framework (Kelley, 1984).

4.2 Framework

Each data point is a fully annotated conversation instance. The annotation involves turn-level utterances, speaker (user/agent) of each turn, turn-level slots, as well as the belief states (which also contains information about the domains under consideration). Instead of looking at each turn separately, we focus our attention to convert each turn-pair (consecutive user-agent turns) in the conversation instance into a node of a graph $\mathcal{G}$. Let $\mathcal{G} = (V, E)$, where V is the set of nodes in the graph $\mathcal{G}$ and E is the set of associated edges. The edge set E is a tuple (u, v) signifying a directed edge, where $u, v \in V$. In our case, each successive turn-pair in a conversation is connected through a directed edge based on the flow of the conversation. To re-iterate, we only use the fully annotated MultiWOZ2.1 training set for constructing this graph, and hence some natural conversation flows might not be present in the graph $\mathcal{G}$.

Each node is represented as a binary vector containing information about the evolution of dialogue/belief-states. Each co-ordinate in the binary vector represents if a specific belief-state-value has been filled (represented using 1) or not (represented using 0). Note that the representation for belief-state being adopted in our model is *agnostic* of the domain being considered. We do this because many domains share common traits which are used interchangeably in other domains. For e.g. (*in majority cases*), for a conversation to look *realistic*, the `area` being considered for a domain like `attraction`, is used in other domains like `taxi` (for either departure or arrival area). We additionally keep two states, namely the *source* and the *sink* states which denote the start and end of a conversation. It should be noted that we do not take into consideration the actual slot values. This provides an abstraction to the conversation flow, for e.g., two conversation might have very different slot values, but can have the same *conversational flow*. This is pictorially depicted in Figure 1.

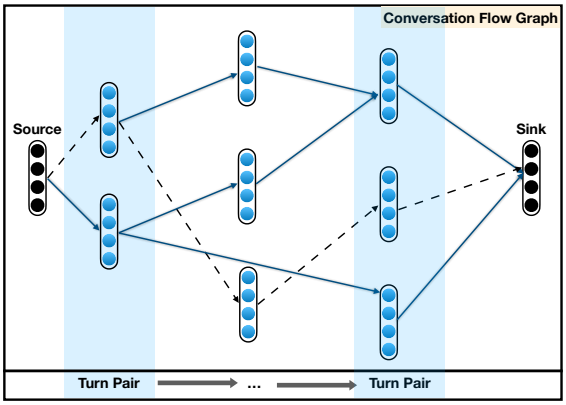

Figure 3: Conversation Flow Graph $\mathcal{G}$: Each node is turn-pair level binary vector containing information about the current belief-state. A path (trace of the dotted line in the figure) represents an abstraction of a data point (conversation) in the data set. When mapped to a natural language each path is essentially an instance of the data set. Please refer Section 4.2 for details

Each data point/conversation instance essentially exists as a path in $\mathcal{G}$ starting from the *source* state and terminating in the *sink*. The nodes alternate between user and agent turns. The graph $\mathcal{G}$ is depicted in Figure 3

4.3 Analysis

Experimental Setup

Having obtained the abstract graph $\mathcal{G}$ from the training set D_{train}, we partition the test set D_{test} into two parts:

1. Non-violation conversation instances (**NV**): Contains those test set instances where the complete conversation exists as a path in the graph.

2. Violation conversation instances (**V**): Contains the complementary of set **NV**. This can be qualitatively thought of as those conversation instance in the test set which deviate from the graph at a certain node i.e., no edge exists in the graph $\mathcal{G}$ from a certain state or turn of the conversation.

In essence, **NV**$\cap$**V** = ϕ

We then assess the joint accuracy scores of each set using the models described, subsequently. Joint Goal Accuracy in DST is the ratio of dialog turns in the data set for which all slots have been filled with the correct value according to the ground truth.

Dialogue State Tracking Models

We measure the joint accuracy scores of the predictions obtained using the following SOTA models:

1. **BERT Based:** TripPy (Heck et al., 2020) is a DST model which uses one of the following three copy mechanisms to generate dialog states (a) Span prediction directly from user utterance, (b) Copy from system-inform memory, (c) Copy from different slot but similar intent, e.g., *area* is one of such slots.

2. **Non-BERT Based:** DSTQA (Zhou and Small, 2019) models a multi-domain DST problem as a question answering problem and leverages dynamic knowledge graph which explicitly learns relationship between multiple (domain, slots) pairs.

Results

Model	Violation	Non-Violation	Overall
TripPy	42.3 (43.2)	65.8 (68.6)	56.0 (59.0)
DSTQA	33.9 (38.2)	56.9 (59.7)	50.8 (54.2)

Table 1: Joint Accuracy[1] results across two models, TripPy and DSTQA. Format: Test (Validation).

The results of the analysis are tabulated in Table 1. We take the mean value of the scores across 5 different checkpoints (last 5). It is evident that the performance on instances which conform to the styles already modeled in the training data i.e., the set **NV**, is higher than instances in the violation set **V**. Upon closer manual inspection of the violation test conversations, we observed some similar trends (subsequence in the conversation - also referred to as *complex patterns*), after which the performance of the DST models declined. These included, but were not limited to, multi-slots being filled in a single turn, turn repetition and speaker asking for clarification. It gave an indication that if we induce those difficult but natural patterns in the test sets, then the resulting test set would be competitive for the current models.

[1]Based on the models trained by us from scratch.

Dataset	Train	Test	Val
MultiWOZ2.1	8438	1000	1000
TaskMaster	17289	NA	NA

Table 2: Dataset statistics

One of the naive ways to approach this infusion of natural patterns is to manually insert them into the current test set (Ganhotra et al., 2020). However, this approach is not only cumbersome but also restricted by ones' imagination. We analyzed other goal-oriented conversational data sets and found that the TaskMaster data set, which was built out of *spoken* conversations, had an abundance of those naturally occurring but difficult patterns. In the next section, we describe the automated approach we took to infuse the natural patterns into the current MultiWOZ2.1 test set.

5 FRAPI: Automated Pattern Infusion

5.1 Datasets

TaskMaster (Byrne et al., 2019) contains single-domain goal-oriented conversational data. In contrast to MultiWOZ2.1, data points in TaskMaster do not contain dialog state annotations but contain more naturally occurring realistic variations in the conversations. This can be attributed to the construction mechanism for Taskmaster. Two collection procedures were adopted for its construction: (a) Two-person *spoken* conversations using the WOZ approach and (b) *self-dialog* where in the complete dialog is constructed using a single crowd-worker. The relevant data set statistics is mentioned in Table 2.

5.2 Modeling

A conversation C consists of multiple alternate turns of user and agent utterances. Most of the current dialog-systems use the following blueprint: User Utterance/Turn (U) $\rightarrow$ Natural Language Understanding (NLU) $\rightarrow$ Dialog-Management (DM) $\rightarrow$ Dialog-policy (DP) $\rightarrow$ Agent Utterance (A).

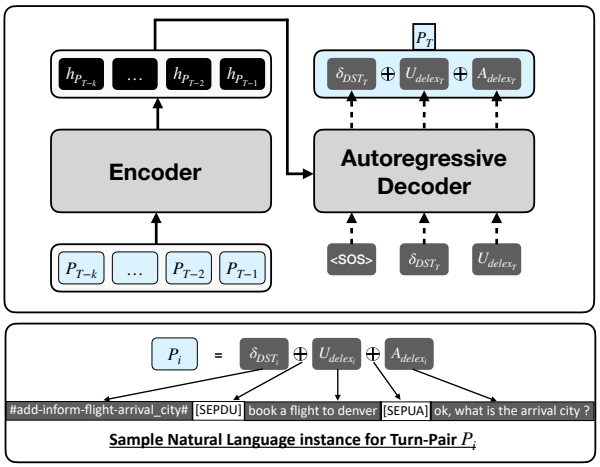

Figure 4: Transformer Network: The encoder takes as input the previous context turn-pair history $\{P_i\}$ and produces the encoder representation $\{h_{P_i}\}$. The encoded representation is utilized by the decoder to produce the current turn-pair P_T. Please refer Section 5.2 for details

Dialog-Management (DM) additionally comprises the database access to fill up the belief states. Instead of thinking about conversations at turn level, we consider turn-pair level conversations i.e., each turn comprises the user utterance as well as the subsequent agent utterance. Intuitively, such a representation makes sense since the belief state changes only during the agent side (and not the user turn). We convert each turn-pair into a tokenized string comprising three main components, each separated by a unique identifier:

1. δ_{DST}: The change in the dialog state from one turn pair to the next.

2. U_{delex}: De-lexicalized version of the user utterance.

3. A_{delex}: De-lexicalized version of the agent utterance.

Compactly represented,

$$D = \{P_i\}_j = \{(\delta_{DST}, U_{delex}, A_{delex})_i\}_j$$

where i is the number of turn pairs (P) in a conversation and j is the number of conversation data points in the set D.

$$\delta_{DST_i} = DST_{i+1} - DST_i$$

$$DST_0 = \phi$$

DST_i is the tokenized representation of dialog states at turn-pair level i and the '$-$' (set subtraction between DST_{i+1} and DST_i) in the equation indicates the change in dialog state from turn-pair i to $i+1$. Note that we maintain a dictionary of delexicalized tokens so that once the conversation is generated, it can be re-lexicalized with necessary values. We additionally augment the representations with and `<eod>` symbol at the end of each conversation, denoting the *end-of-dialog*. An example of this can be found in Figure 4.

Since our main aim is to generate conversation test points along with the dialog states, we rely on the transformer network (Vaswani et al., 2017). Transformer network is an encoder-decoder model (Sutskever et al., 2014) which uses self-attention to encode the context history (previous 4 turn-pairs) of a conversation flow and outputs the current turn-pair in an auto-regressive manner via the decoder. The network comprises 6 layers of multi-head self-attention network each in the encoder as well as the decoder.

5.3 Experimental Setup

We train the transformer network on the accumulated training data set: MultiWOZ2.1 + TaskMaster. Since TaskMaster does not contain annotated dialog states, we treat change in sentence level slot labels over subsequent turns as a proxy for δ_{DST}. Once we accumulate the entire training data set, we train the transformer network as described earlier in Section 5.2.

We then use the trained model to generate competitive test samples. The prompt given to the transformer network (initial state) is the first two turn-pairs of each data point in the MultiWOZ2.1 test set. Based on the initial prompt, the transformer generates subsequent turn-pairs. We accumulate the generated turn-pair to the previous turns for generation of next turn-pairs. Since the model generates $\{\delta_{DST_T}, U_{delex_T}, A_{delex_T}\}$, we first separate δ_{DST_T} from the generation and then with probability p, give the inference model a choice to either replace the generated δ_{DST_T} with `None` value or keep it as is. The reason for `None` is that for patterns like repetition and clarification, dialogue state does not change, hence it makes sense to use $\delta_{DST_T} =$ `None` for those turns. The control over p helps in inculcating diversity or *complex patterns* into the final test point. We have the choice of introducing human intervention during the decoding process for sanity check. This is especially useful, in case the model outputs unwanted tokens, akin to CHECKLIST. The process stops when the model

Prompt/Generated	Turn	Conversation
PROMPT	1	are there any catalan restaurants in the centre of town ?
	2	i 'm sorry , there are no catalan dining establishments in the centre . would you like to look for a different cuisine or area ?
GENERATED	3	are there any european restaurants in the centre ?
	4	i 'm sorry , there are no european restaurants in the centre. would you like to try another area ?
	5	how about american food ?
	6	there are 9 american restaurants in the centre . i recommend bar . would you like a reservation ?
	7	*yes , please book a table for 2 people at 20:00 on sunday .*
	8	*what time would you like to dine ?*
	9	*i would like the reservation for 20:00 please .*
	10	booking was successful . the table will be reserved for 15 minutes . reference number is : fucdlrg3 . is there anything else i can help you with ?
	11	no , that 's all i need . thank you for your help !
	12	you 're welcome ! have a great day !

Table 3: Generated example showing the incorporation of repetition and clarification in the MultiWOZ2.1 test set. We only provide the first two turns as the prompt to the Transformer Network, and the rest of the turns (turn-pairs) are generated through the decoder. Note that white cells are for user turns and gray cells are for agent turns. Please refer to Section 6.1 for details.

either outputs the <eod> symbol or the maximum allowable generation turn-pairs are exhausted.

Why does it make sense to use Dialogue States and not Dialog Act in the representation

Dialog act is a by-product of the user or agent utterance itself. The labels associated in dialog acts are essentially tokens already present in the user/agent utterances. Since, we are using utterance tokens as an input to the encoder, providing that information again to the encoder in another form would only increase redundancy. On the other hand, dialogue state is a by-product of dialogue acts as well as the database queries issued by the agent. The access to the database provides some vital details to the dialogue state that are not *directly* encoded in the previous utterances.

5.4 Implementation Details

We use sockeye (Hieber et al., 2017) implementation of 6-layered transformer networks for generating turn-pairs. Each layer in the transformer block is composed of 8 headed multi-head attention network with residual connections.We use the Adam optimizer (Kingma and Ba, 2014) with $\beta = 0.9, 0.999$ and an initial learning rate of $1e-4$ with warm-up step size of 4000. The network typically reaches convergence around 25 epochs. We train our model on four Nvidia V100 GPUs. During inference, we set $p = 0.3$ and the maximum number of generated turn pairs are restricted to 10. Setting higher values of p results in more number of *difficult patterns*, which might not always be desirable.

5.5 Evaluation Protocol

Once the conversations are generated, we measure the performance of the models using the joint accuracy scores. Additionally, we assess the quality of generations through human evaluation. Human evaluation involves assessment of the conversation sets by crowd-workers. The crowd-workers provide score on a **5-level** Likert scale (1 being the lowest and 5 being the highest) for (a) coherence: whether the conversation follows a logical and natural order, (b) believability: whether the conversation look like a realistic conversation between an agent and a user and a score on a **3-level** Likert scale for (c) appropriateness: does the user/agent ask appropriate questions on the conversation topic.

6 Results

6.1 Qualitative Analysis

In utterance-level conversation generated by the transformer network as highlighted in Table 3, it can be observed that patterns like repetition and clarification are naturally infused into the generated set. A particularly interesting example of this complex pattern is from turn 7-9 (italicized text). We can see that the user has indicated the booking time as 20:00 hrs, but the agent is not able to comprehend that in the first go. It is easy to imagine this scenario occurring in realistic conversations where the agent is unable to comprehend the time because of various issues, one of which can be difficulty in understanding the dialect of the user. These types of naturally occurring patterns are generally missing in the current test set. It should be noted that the transformer generates only the delexicalized strings during inference. They undergo

Repetition	Clarification	Multi-value State info.
...	...	U: i 'm looking for an attraction called williams art and antiques .
A: how many people will be staying ?	U: i would like to travel on friday .	A: williams art and antiques is at gwydir street , no . 5 dale 's brewery . would you like me to book tickets ?
U:i would like it for 4 nights	A: i 'm sorry , i do n't have anything leaving at that time .	U: no thank you . just the area and postcode please
A: how many people will be staying ?	U: i would like it to be on friday at 10:00 .	A: it is in the 4017 hillsboro pike #301, nashville, tn 37215 and the postcode is cb12lj . is there anything else you need help with today ?
U: i would like a booking for two people for 4 nights, starting on saturday	A: i 'm sorry , i was n't able to find any trains that meet that criteria .	U: i 'm looking for places to go in town
...	...	A: what type of place would you like to visit ?
...	...	U: i am looking for something fun to do .
...	...	A: how about castle galleries ?
...	...	...

Table 4: Sample patterns induced by FRAPI into the conversations. For multi-value state info., the `attraction-name` slot is filled multiple times.

a simple post-processing where the generation is re-lexicalized using the saved dictionary as mentioned earlier. Additional samples in Table 4.

6.2 Human Evaluation

Test Set	Coherence	Believability	Appro.
Original	4.4	4.4	2.8
MTASK-TEST	4.1	3.9	2.5

Table 5: Human evaluation results on the original test-bed and the generated conversations. Please refer to Section 6.2 for details.

We randomly sample 50 test set conversations from the current MultiWOZ2.1 data set as well as 50 conversations generated by the Transformer model (Vaswani et al., 2017) with *human in the loop* setup. For this setup, we allow at max one correction (addition/removal of an utterance) per conversation. The sampled conversations are then sent to crowd-workers for assessment on the following three qualities: (a) coherence, (b) believability, (c) appropriateness, as described in Section 5.5. Table 5 contains the compilation of those results. We observe that the generations with minimal human intervention have competitive quality as the original test-bed. In addition, FRAPI took approximate 22 seconds per conversation (mostly validation) compared to the pilot human-human collection that took an average of 137 seconds per conversation collection, thus leading to significant reduction of burden on data collection while introducing the nuances with *complex patterns*.

6.3 DST Task Performance

The evaluation results of the test set on current baseline models Section 4.3 are highlighted in Table 6. We can see that the joint accuracy performance of the models fall drastically on the generated test set, as compared to the current test set. This indicates that current models are not able to account for realistic variations which are omnipresent in natural conversations and cling onto perfect signals obtained from restricted test sets. A possible cause might be because the same set of biases are present in the test set as in the training set.

Model	Original Test Set	MTASK-TEST
TripPy	56.01	27.12
DSTQA	50.78	25.76

Table 6: Comparison of performance of current baseline models on the original test set against the proposed test set

7 Conclusion

There are umpteen ways in which a task-oriented conversation might proceed. This variability only increases with the induction of more domains into the conversations. We analyzed the limitations of diversity in current test beds and the importance of inducing diversity at conversational flow level using a graphical abstraction. Based on our findings, we proposed FRAPI, a framework for inducing rich conversation patterns into current test sets through Transformer Networks. The resultant test dialogues were found to be challenging for current systems. While we provide a general framework for generating good test sets, we anticipate better assistive capabilities and results using pre-trained seq2seq models like BART (Lewis et al., 2020). Given the simplicity of this approach, we believe that will be helpful in building competitive test beds for other dialogue data sets.

References

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

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)*.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4517.

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*.

Jatin Ganhotra, Robert C Moore, Sachindra Joshi, and Kahini Wadhawan. 2020. Effects of naturalistic variation in goal-oriented dialog. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4013–4020.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Milan Gritta, Gerasimos Lampouras, and Ignacio Iacobacci. 2020. Conversation graph: Data augmentation, training and evaluation for non-deterministic dialogue management.

Arshit Gupta, AI Amazon, John Hewitt, and Katrin Kirchhoff. 2019. Simple, fast, accurate intent classification and slot labeling for goal-oriented dialogue systems. In *20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 46.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

John F Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3609–3619.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *Interspeech 2016*, pages 685–689.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *arXiv preprint arXiv:2009.13570*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 3104–3112. MIT Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *AAAI Conference on Artificial Intelligence*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018b. A bi-model based rnn semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 309–314.

Jason D Williams, Pascal Poupart, and Steve Young. 2005. Factored partially observable markov decision processes for dialogue management. In *Proc. IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 76–82.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019a. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019b. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *arXiv preprint arXiv:1911.06192*.