End-to-End Learning of Flowchart Grounded Task-Oriented Dialogs

Dinesh Raghu [* † 1 2]**, Shantanu Agarwal** [* 1]**, Sachindra Joshi** [2] **and Mausam** [1]

[1] Indian Institute of Technology, New Delhi, India
[2] IBM Research, New Delhi, India

```
diraghu1@in.ibm.com, a.shantanu08@gmail.com,
jsachind@in.ibm.com, mausam@cse.iitd.ac.in
```

Abstract

We propose a novel problem within end-to-end learning of task oriented dialogs (TOD), in which the dialog system mimics a troubleshooting agent who helps a user by diagnosing their problem (e.g., car not starting). Such dialogs are grounded in domain-specific flowcharts, which the agent is supposed to follow during the conversation. Our task exposes novel technical challenges for neural TOD, such as grounding an utterance to the flowchart without explicit annotation, referring to additional manual pages when user asks a clarification question, and ability to follow unseen flowcharts at test time. We release a dataset (FLODIAL) consisting of 2,738 dialogs grounded on 12 different troubleshooting flowcharts. We also design a neural model, FLONET, which uses a retrieval-augmented generation architecture to train the dialog agent. Our experiments find that FLONET can do zero-shot transfer to unseen flowcharts, and sets a strong baseline for future research.

1 Introduction

Task oriented dialog (TOD) systems (Bordes and Weston, 2017) converse with users to help them with specific tasks such as calendar enquiry (Eric et al., 2017), restaurant reservation (Henderson et al., 2014), and tourist package recommendation (El Asri et al., 2017). These dialog systems (e.g., restaurant reservation system) are trained using past human-to-human dialogs and associated knowledge sources (e.g., a KB of restaurants).

Most existing TOD systems are conversational *recommender* systems that gather user requirements in the form of attributes (such as cuisine, location), query a KB and generate recommendations based on the retrieved results (e.g, restaurant, its phone number). While there have been recent

Table 1: Type of utterances and their proportions in a real-world troubleshooting dialog dataset.

Utterance Type	%
[T1] Problem Description	6.09
[T2] Grounded on Flowchart	56.76
[T3] Grounded on Supplementary Knowledge	15.23
[T4] Chit-Chat	5.23
[T5] Conversation Markers	7.62
[T6] Hold Request	2.38
[T7] Reconfirmation	0.95
[T8] Dialog Closing	5.71

efforts (Feng et al., 2020; Kim et al., 2020) to study non-recommendation TOD, several important tasks such as troubleshooting are still unexplored.

Troubleshooting is a common task handled by customer support agents. It involves understanding a user's problem, narrowing down the root cause and providing a solution. Figure 1 shows an example dialog between an agent and a user troubleshooting a car problem. Support agents typically follow a flowchart (utterances A1, A4 in our example) to diagnose user problems, but may refer to supplementary knowledge sources like FAQs (A3), if user asks a clarification question (U3).

In this paper, we propose the novel task of end-to-end learning of a TOD system that troubleshoots user's problems by using a flowchart and a corpus of FAQs. Our task exposes novel research challenges for TOD system design. First, the system must learn to ground each utterance in the flowchart without explicit supervision. Second, when required, the agent must refer to additional knowledge in the corpus of FAQs to issue clarifications and add details not present in the flowchart. Third, it must learn the general skill of following a flowchart, which is tested in a zero-shot transfer setting with unseen flowcharts shown at test time.

Before collecting a dataset for the task, we first analyze a sample of 100 in-house troubleshooting dialogs with a human customer service agent. Table
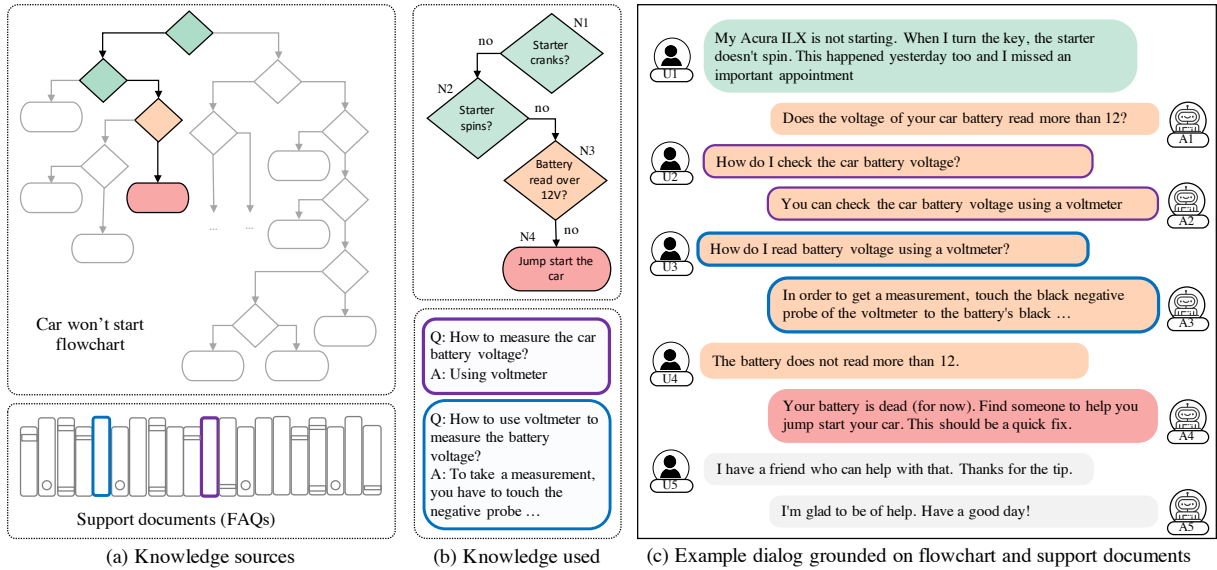

Figure 1: Example flowchart grounded TOD. It is grounded on two knowledge sources: flowchart and FAQs.

1 summarizes the statistics on common utterances in such conversations. This analysis reaffirms the importance of supplementary knowledge (T3).

We crowdsource the first version of our dataset, FLODIAL[3] (**Flo**wchart Grounded **Dial**ogs), with these utterance types: problem description (T1), flowchart following (T2), use of supplementary knowledge in the form of FAQs (T3), and closing utterances (T8). FLODIAL has 2,738 dialogs grounded on 12 different flowcharts.

Since this is a new task, existing end-to-end TOD models are not directly applicable to it. We design a baseline network named FLONET[4] – it follows the retrieval augmented generation framework (Lewis et al., 2020) and generates agent response in two steps. First, relevant information from the flowchart and FAQ corpus is retrieved based on the dialog history. Then, this retrieved information and dialog history generate the agent response using an encoder-decoder. We evaluate FLONET in two different settings: (1) *Seen Flowcharts* (S-Flo) setting, tested on flowcharts seen at train time, and (2) *Unseen Flowcharts* (U-Flo) setting, to evaluate FLONET's zero-shot transfer ability in handling new flowcharts unseen at train time. To summarize, the main contributions of this paper are:

1. We propose the novel problem of end-to-end learning of flowchart grounded task oriented dialog.
2. We collect a new flowchart grounded task-

oriented dialog (FLODIAL) dataset.
3. We propose a baseline solution (FLONET) for the proposed problem and evaluate it in *seen flowchart* and *unseen flowchart* settings.

We release all our resources for further research on the task.

2 Related Work

Dialog systems can be broadly divided into two types: task oriented (TOD) (Williams and Young, 2007; Bordes and Weston, 2017) and open domain dialog systems (Vinyals and Le, 2015; Serban et al., 2016). Task oriented dialogs systems can further be divided into end-to-end (Bordes and Weston, 2017; Raghu et al., 2019, 2021; Gangi Reddy et al., 2019) and traditional slot filling approaches (Williams and Young, 2007). Slot filling approaches require dialog state annotations in dialog transcripts. Our work falls under end-to-end approaches, which do not require any such intermediate annotations. We first briefly discuss existing TOD datasets and then review approaches for collecting dialog datasets. Finally, we discuss dialog systems related to FLONET.

Dialog Datasets: Exisiting TOD datasets can be grouped based on the type of knowledge source on which the dialogs are grounded. Most of the existing datasets are for the recommendation task and grounded on structured KBs. Some notable KB-grounded datasets are MultiWOZ (Budzianowski et al., 2018), Stanford multi domain dataset (Eric et al., 2017), CamRest (Wen et al., 2016), Frames

(El Asri et al., 2017), schema guided dialogs (Rastogi et al., 2020) and taskmaster-1 (Byrne et al., 2019). Kim et al. (2020) augment MultiWOZ with utterances grounded on FAQs. The dialogs in datasets such as ShARC (Saeidi et al., 2018) and doc2dial (Feng et al., 2020) are grounded on snippets from unstructured text documents. To the best of our knowledge, FLODIAL is the first TOD dataset that is grounded on flowcharts and FAQs.

Dialog Data Collection: Crowd sourcing frameworks for creating dialog datasets can be broadly grouped into three types. (1) Wizard-of-Oz framework (Kelley, 1984) pairs up two crowd-workers who play the roles of user and agent while conversing. The user is provided with a goal and the agent is given the knowledge necessary to achieve the goal. (2) Self-dialogs framework (Byrne et al., 2019) requires a single crowd-worker to write the entire dialog by playing both user and agent. (3) Dialog paraphrasing framework (Shah et al., 2018) systematically generates a dialog outline (user and agent utterance) and crowdsources paraphrases for each utterance to construct a dialog. We follow this framework for collecting FLODIAL, as it gives us adequate control over dialog flow so that we can incorporate various utterance types in Table 1.

Dialog Systems: Large scale pre-trained language models such as GPT2 (Radford et al., 2019) have been used for response generation in both open domain (Wolf et al., 2019; Zhang et al., 2020; Zhao et al., 2020) and TOD systems (Ham et al., 2020; Hosseini-Asl et al., 2020). A major challenge is GPT2's limitation on the input size. For our setting, it becomes difficult to feed a long input (flowchart, dialog history, FAQ corpus) to GPT2. We overcome this by following the retrieval augment generation paradigm (Lewis et al., 2020) – we are probably the first to apply it to a dialog setting.

The task of zero-shot response generation requires a model to generalize to new domains with just domain descriptors and no training dialogs. Existing approaches (Zhao and Eskenazi, 2018; Wu et al., 2019; Rastogi et al., 2020) model slots and intents as domain descriptors. We model flowcharts as domain descriptors and expect the system to generalize to new flowcharts unseen during train.

3 The FLODIAL Dataset

FLODIAL is a corpus of troubleshooting dialogs between a user and an agent collected using Amazon Mechanical Turk (AMT). The dataset is accompanied with two knowledge sources over which the dialogs are grounded: (1) a set of troubleshooting flowcharts and (2) a set of FAQs which contains supplementary information about the domain not present in the flowchart – both are in English.

The data collection process uses the dialog paraphrasing framework (Shah et al., 2018) and is illustrated in Figure 2. At a high level, we first systematically construct an outline for each dialog, then decompose the outline into multiple AMT paraphrasing tasks, and finally stitch the dialog using the collected paraphrases. Our data collection process has the following advantages: (i) systematic outline construction guarantees coverage of all paths in the flowchart, and the desired balance of utterance types in dialogs, (ii) the process ensures the annotated labels[5] are always correct and (iii) it provides diversity in the paraphrases collected.

3.1 Flowcharts and FAQs

We identify 12 flowcharts[6] on troubleshooting laptop and cars problems, such as overheating laptop, car won't start and car brake failure. The flowcharts encode agent questions as decision nodes and user responses as edges. The agent follows flowcharts based on user responses to reach a terminal node (e.g., node N4 in Figure 1b) which contains the solution. We refer to the sequence of nodes and edges from root to a terminal node as a path in the flowchart. One such path is shown in Figure 1b.

The flowcharts usually contains precise instructions with no details. For example, node N4 in Figure 1b just says "does the battery read over 12V?" but does not provide details such as "which instrument is needed to measure the battery voltage?" or "how does one measure the battery voltage using a voltmeter?". For each flowchart, we collect supplementary FAQs[7] that contain details such as step-by-step instructions for a process (e.g., "how to jump start your car?") and other common doubts (e.g., "where is the ignition coil located?"). A few example FAQs are shown in Figure 1b.

3.2 Dialog Outline Construction

We systematically iterate over paths in the flowchart and for each path we construct multiple outlines. Each outline consists of 3 major parts:

---

[5]We do not use these annotated labels during train, but use them to evaluate the performance of the dialog system.
[6]Downloaded with permission from www.ifitjams.com
[7]Collected in-house, refer Appendix A.7

| (a) Dialog Outline | (b) AMT Paraphrasing Tasks | (c) Dialog Constructed using Collected Paraphrases |

Figure 2: An example of a dialog outline, AMT task creation and paraphrasing. Each dotted line box denotes a single component of the dialog. These components are independently paraphrased and then finally stitched together to construct one dialog. Each colored bubble in (b) denotes an AMT task and the matching bubble in (c) denotes the corresponding collected paraphrase. The last two paraphrases in (c) are for the closing task in (b). Paraphrases from non-contextual tasks are used in the corresponding contextual tasks, as denoted by the arrows.

problem description, flowchart path traversal and closing. We now discuss each part in detail.

Problem Description: The problem description is the first utterance in a dialog. It contains (1) the primary issue faced by the user, (2) secondary information, and (3) other information that may not be relevant for troubleshooting. The primary issue is phrased using title of the flowchart. For example, for Figure 2 it will be *car won't start*. The secondary information is any other information that may help in troubleshooting the primary issue. For example, the user could say that *starter is not cranking*. This secondary information is populated by sampling a random (node, edge) pair from the sampled flowchart path. For example, (N1, no) is populated as a secondary information in Figure 2. By adding this to problem description, we mimic the setting where, an agent may need to *skip* a few nodes when following the flowchart, based on information already present in the dialog history.

Flowchart Path Traversal: After the problem description, we consider each (node, edge) pair in the given flowchart path. For each (node, edge) pair we toss a coin to decide if the pair should be represented as a simple exchange or as a complex exchange. A simple exchange is one where the

agent asks the question in the node and the user responds with the answer in the edge. (N2, *No*) in Figure 2c is constructed as a simple exchange. Complex exchanges use at least four utterances to represent the information in the (node, edge) pair, e.g., (N3, *No*) in Figure 2c. Complex exchange can be of two types: user-initiated digression and agent digression. The example illustrates *user digression* where the user asks for clarifications to understand the agent question before responding with an answer. An *agent digression* is similar except that the agent proactively breaks a complex question into a sequence of simple ones. An example agent digression for (N3, *No*) would be when the agent first asks "Do you know how to measure the voltage of a car battery using a voltmeter?". If the user responds "no", the agent will then describe the procedure to measure the voltage, and then requests the user to check if the voltage is greater than 12V.

Closing: Closing contains the solution suggested by the agent followed by one or more exchanges to gracefully terminate the dialog. Typically, the users thank the agent and the agent terminates the dialog by acknowledging the user.

3.3 AMT Paraphrasing Tasks

We crowdsource paraphrases of each utterance in a dialog outline. Utterances corresponding to each component (problem description, node-edge pairs in the flowchart path and closing) are paraphrased separately and then stitched together to construct a dialog. We define four types of paraphrasing tasks: non-contextual, contextual, problem description and closing tasks. In the non-contextual task, a single utterance from the outline is provided to the crowd workers to paraphrase. We requested the workers to provide two paraphrases for each utterance to improve diversity among paraphrases (Jiang et al., 2017; Yaghoub-Zadeh-Fard et al., 2019). In the contextual task, workers are asked to paraphrase in the context of a specific previously collected paraphrase. Problem descriptions tasks ask the worker to describe the troubleshooting problem using the primary issue and secondary issue as discussed in Section 3.2. In closing task, the worker gracefully terminates the dialog in the context of a troubleshooting solution collected from a non-contextual task. Examples of the four type of tasks can be seen in Figure 2b.

As most user responses in a flowchart are yes/no, we design the yes/no paraphrasing task based on a study by Rossen-Knill et al. (1997). We add specific rules in the tasks for workers to follow when paraphrasing a yes/no user response. An example (in blue outline) is shown in Figure 2b.

3.4 Dialog Construction

We generate around 110 outlines for each flowchart by equally dividing them amongst the paths in the flowchart. We generate a total of 1,369 outlines and then collect paraphrases of the constructed outline components. Finally the component paraphrases are stitched together to construct 1,369 dialogs as shown in Figure 2c.

The paraphrases corresponding to an outline component are interchangeable across dialogs. We take advantage of this and generate an additional 1,369 dialogs by randomly interchanging paraphrases without breaking semantics. Our final set has 2,738 dialogs with an avg of 15.56 utterances per dialog. The agent and user utterances have an average of 14.95 and 16.17 words in them.

3.5 Paraphrase Cleaning

To avoid error propagation, we manually verify all paraphrases and correct errors in grammar, spelling and polarity. It took an author approximately 5 minutes per dialog for this step. An example of a polarity error is when the question '*Do you have an open circuit?*' was paraphrased as '*Do you have a closed circuit?*' by a crowd worker. Such paraphrases invert the semantics of (yes/no) edges from the given node and will break the correctness of a dialog, if not corrected. About 6% utterances were recollected as they violated instructions.

4 Task Definition & Baseline System

In this section, we define the problem of learning flowchart grounded task oriented dialogs in an end-to-end manner without the use of intermediate labels. We then describe our proposed baseline model, FLONET, which retrieves necessary knowledge from flowchart/FAQs and generates the agent response using the retrieved knowledge.

4.1 Task Definition

We represent a dialog d between a user u and an agent a as a sequence of utterances $\{c_1^u, c_1^a, c_2^u, c_2^a, \ldots, c_m^u, c_m^a\}$, where m denotes the number of exchanges in the dialog. Let $\mathcal{F} = (N, E)$ be the flowchart over which the dialog d is grounded, where the set of nodes N represents the agent questions and edges E represent the user responses. The number of outgoing edges from a node depends on the number of possible user responses for the agent question associated with the node. Let $\mathcal{Q} = \{q_l : a_l\}_{l=1}^L$ be the set of frequently asked question and answer pairs (FAQs) associated with the flowchart $\mathcal{F}$. Our objective is to learn a next response predictor, which takes (1) the dialog-history $\mathbf{h} = \{c_1^u, c_1^a, \ldots, c_i^u\}$, (2) a flowchart ($\mathcal{F}$), and (3) a set of FAQs ($\mathcal{Q}$) as input and predicts the next agent response ($\mathbf{y} = c_i^a = \langle y_1 y_2 \ldots y_T \rangle$).

4.2 Baseline System: FLONET

Since it is a novel task, an existing TOD architecture does not directly apply on this problem. We design a baseline architecture named FLONET for predicting the agent responses in flowchart grounded dialogs. FLONET is trained in an end-to-end manner without the need for any intermediate annotations such as (a) whether the given agent utterance is grounded on a flowchart node or FAQs, or (b) the specific flowchart node or FAQ on which the agent utterance is grounded.

FLONET follows the retrieval augmented generation framework (RAG) (Lewis et al., 2020; Guu

et al., 2020), which first retrieves the necessary knowledge to generate the response and then generates the agent response one word at a time by using the retrieved knowledge. The framework consists of two main components, a retriever and a generator. The retriever $p_\eta(z|\mathbf{h})$ outputs a distribution over all documents based on the dialog history $\mathbf{h}$. The flowchart $\mathcal{F}$ and FAQs $\mathcal{Q}$ are represented as documents (discussed further in Section 4.2.1). The generator $p_\theta(y_t|\mathbf{h}, z, y_{1:t-1})$ generates the agent response y word by word by using the dialog history $\mathbf{h}$ and a retrieved document z. We generate the response using RAG-Sequence model:

$$p(\mathbf{y}|\mathbf{h}) = \sum_{z \in N \cup \mathcal{Q}} p_\eta(z|\mathbf{h}) \prod_{t=1}^{T} p_\theta(y_t|\mathbf{h}, z, y_{1:t-1}) \quad (1)$$

The overall network is trained by minimizing the negative log-likelihood of the response given by Equation 1. Following Lewis et al. (2020), we marginalize over all the documents using a top-k approximation. We use top-5 documents in our training implementation due to memory constraints. During inference, only the top-1 document is used because the dialog's agent responses need to be grounded on only one flowchart node or FAQ. This is unlike RAG where multiple documents extracted from Wikipedia can contribute to the expected output. See Appendix A.3 for further details.

4.2.1 Retrievable Documents

The retrievable document set includes all flowchart nodes and all FAQ QA pairs associated with the flowchart. In the original RAG model, each (Wikipedia) document had a single dense embedding, based on which a document was retrieved and used. However, for our setting, the content of a flowchart node will typically not be explicitly mentioned in the dialog history. Instead, the right node is best determined based on the flowchart structure – the path to that node – as expressed in the dialog history. Similarly, for FAQs, a QA-pair will typically be matched on the question and the answer will be used in subsequent dialog.

Consequently, we represent each document as a key-value pair. The document-key is used by the retriever to compute $p_\eta(z|\mathbf{h})$ and the document-value is used by the generator during response generation. We construct a document for each node in $\mathcal{F}$ and for each FAQ in $\mathcal{Q}$. The document-key of a flowchart node is the sequence of utterances corresponding to the nodes and edges in the path from the root. Its document-value is the agent utterance associated with it. For a FAQ, the document-key and value are the question and answer, respectively.

4.2.2 Retriever & Generator

The retriever scores each document z based on the dialog history. The dialog history is encoded using a hierarchical recurrent encoder (Sordoni et al., 2015). The encoder computes a dense representation of the history $\phi_h(\mathbf{h})$. The document-key is also encoded using a hierarchical recurrent encoder to compute its vector representation $\phi_z(\mathbf{z})$. For each document, we assign a score as negative of the Euclidean distance between $\phi_h(\mathbf{h})$ and $\phi_z(\mathbf{z})$. The top-k scores are then passed through a Softmax layer to compute $p_\eta(z|\mathbf{h})$. We use GPT2 as the generator $p_\theta(\mathbf{y}|\mathbf{h}, z)$ and it receives a separate input for each retrieved document z. The input to GPT2 is constructed by concatenating all the utterances in the dialog history along with the document-value. GPT2 input is described in detail in Appendix A.2. The response is decoded using beam search.

4.2.3 Pre-training

To provide a good initialization to the retriever and the generator, we pre-train both the components separately. For each dialog history and response pair $(\mathbf{h}, \mathbf{y})$ in our dataset, we first identify the document over which the response is grounded using weak supervision (Zhao et al., 2020). The document whose document-value has the highest BLEU score (Papineni et al., 2002) w.r.t. the response $\mathbf{y}$ is labeled as the pseudo grounded document.

The retriever is pre-trained using a contrastive loss (Hadsell et al., 2006) by using the pseudo grounded document as the positive example and any other random document as a negative example. The generator is pre-trained by minimizing the negative log likelihood of the response given the dialog history and the document-value of the pseudo grounded document. Following Wolf et al. (2019), we add a next-utterance classification loss to the negative log likelihood loss. The classification loss is applied on the output of a linear classification layer which receives the last hidden state of the generator and outputs the probability of a given utterance being the correct agent response. We use randomly sampled incorrect utterances as negative examples to train the generator based classifier.

	S-Flo	U-Flo
Train Dialogs	1,798	1,786
Val Dialogs	456	454
Test Dialogs	484	498

Table 2: Statistics of the dataset split.

Model	S-Flo		U-Flo	
	BLEU	PPL	BLEU	PPL
TF-IDF + GPT2	11.97	12.88	6.45	16.38
FLONET (No PT)	18.90	3.86	14.19	5.35
FLONET	**19.46**	**3.79**	**16.31**	**4.94**
Oracle Ret. + GPT2	23.73	-	24.85	-

Table 3: Next response prediction performance.

5 Experimental Setup & Results

5.1 Data Split

We create two different splits of the dialogs in FLO-DIAL. The *S-Flo* split is used for evaluating the ability of FLONET to generate responses by following flowchart and FAQs. The *U-Flo* split is used to study the ability of FLONET to generalize to flowcharts unseen during train in a zero-shot flowchart grounded response generation setting.

To generate the *S-Flo* split, we divided the dialogs associated with each flowchart as follows: 66% for train set, 17% for validation set and 17% for test set. We randomly select a path in the flowchart and push all the dialogs that follow the path to one set. To generate the *U-Flo* split, we group all dialogs associated with 8 flowcharts as train set, all dialogs from 2 flowcharts as validation set and the remaining 2 into test set. Thus, the U-Flo split has mutually exclusive sets of flowcharts in each set. Some statistics on the dataset split are shown in Table 2.

5.2 Evaluation Metrics

We measure the ability to generate responses using two standard metrics: BLEU and perplexity. As FLODIAL contains the labels of the document (flowchart node or FAQ) over which each agent response is grounded on, we use recall@1 (R@1) to measure the retriever performance. We also compute a task-specific metric called *success rate* (SR) which is measured as the fraction of dialogs for which an algorithm retrieved the correct flowchart-node/FAQ for all the agent utterances in the dialog.

We perform a human evaluation for the responses generated by FLONET and 3 other variants

Model	S-Flo		U-Flo	
	R@1	SR	R@1	SR
TF-IDF + GPT2	0.334	0.002	0.394	0.004
FLONET (No PT)	0.768	0.260	0.586	0.064
FLONET	**0.814**	**0.337**	**0.661**	**0.125**

Table 4: Retriever performance of various models.

of FLONET along two dimensions: (i) *relevance* – the ability to generate responses that are relevant to the dialog context, and (ii) *grammar* – ability to generate grammatically correct and fluent responses. Both the dimensions are evaluated on a Likert scale (0-4) (Likert, 1932).

5.3 Implementation Details

The models were implemented using PyTorch (Paszke et al., 2019). We identify hyper-parameters using a grid-search and identified the best hyper-parameters based on the evaluation of the held-out validation sets. Each hyper-parameter combination was run ten times. We sample word embedding size from $\{50, 100, 200, 300\}$, retriever learning rates (lr_R)[8] from $\{1E\text{-}2, 5E\text{-}3, 1E\text{-}3, 5E\text{-}4, 1E\text{-}4, 5E\text{-}5, 1E\text{-}5\}$, generator learning rates (lr_G) from $\{6.25E\text{-}4, 2.5E\text{-}4, 6.25E\text{-}5, 2.5E\text{-}5, 6.25E\text{-}6, 2.5E\text{-}6, 6.5E\text{-}7, 2.5E\text{-}7\}$, and dropout from increments of 0.02 between [0, 0.2]. Hidden size of the retriever was set to three times the word embedding size in all the settings. The word embeddings of the retriever were initialized with pre-trained GloVe embeddings (Pennington et al., 2014). The generator was built on top code made available by Wolf et al. (2019).[9] The best hyper-parameter settings and other details are in Appendix A.1

5.4 Results

We report the performance of our baseline FLONET on both S-Flo and U-Flo splits of FLODIAL. We also report the numbers for two simple variants of FLONET: TF-IDF + GPT2 and FLONET (No PT). The former variant uses a simple TF-IDF technique to retrieve documents. The top retrieved document concatenated with the dialog history is fed as input to GPT2 for generating a response. FLONET (No PT) is FLONET without the pre-training described in Section 4.2.3.

Table 3 reports the response prediction performance of various systems on both data splits and

---

[8] $a\mathrm{E}b$ denotes $a \times 10^b$

[9] https://github.com/huggingface/transfer-learning-conv-ai

Model	S-Flo		U-Flo	
	Rel.	Gra.	Rel.	Gra.
TF-IDF + GPT2	2.63	3.59	1.13	3.24
FLONET (No PT)	3.11	3.11	2.37	**3.62**
FLONET	**3.12**	**3.46**	**2.55**	2.71
Oracle Ret. + GPT2	3.53	3.65	3.69	3.76

Table 5: Human evaluation of various models.

Data Source	S-Flo		U-Flo	
	BLEU	PPL	BLEU	PPL
DH	11.86	14.64	3.00	19.40
DH + FC	16.91	4.10	13.42	6.45
DH + FC + FAQ	**19.46**	**3.79**	**16.31**	**4.94**

Table 6: Response prediction performance of FLONET with different knowledge sources. DH and FC indicates dialog history and flowchart respectively.

Error Type	% Error (Count)	
	S-Flo	U-Flo
Retrieved Sibling	66.8 (248)	40.6 (348)
Retrieved Parent	2.2 (8)	7.4 (64)
Retrieved FAQ	0.8 (3)	6.5 (56)
Retrieved Other Nodes	30.2 (112)	45.5 (390)

Table 7: Retriever errors (%) on utterances grounded on flowcharts. Error counts are in parentheses.

Digression Type	S-Flo		U-Flo	
	BLEU	R@1	BLEU	R@1
User Digression	22.58	0.77	19.31	0.66
Agent Digression	18.09	0.23	8.24	0.09

Table 8: Performance of the generator (BLEU) and the retriever (R@1) on the utterances grounded on FAQs.

Table 4 reports the performance of the respective retrievers. TF-IDF + GPT2 has reasonable response prediction performance on S-Flo setting, but has a poor U-Flo performance. The poor generalization is due to the TF-IDF retriever's low R@1. This forces the generator to memorize the knowledge necessary to generate a response, rather than inferring it from the retrieved documents.

FLONET achieves a marginal improvement over the No PT variant on S-Flo, and a two point jump in BLEU in U-Flo setting. This shows that the heuristic pre-training contributes to the overall system performance of FLONET. The success rate of various systems is reported in Table 4. The success rate achieved by FLONET retriever in both settings are quite low. We hope this gets improved by further research on the dataset.

The oracle ret. + GPT2 in Table 3 is approximated by assuming a perfect retriever and training GPT2 with ground truth document. The gap in BLEU represents the value of annotation for our task, and the performance gain a better retriever may help achieve.

We also compare the performance of FLONET on the two data splits. We find that while numbers are understandably worse in the U-Flo setting, the zero-shot transferred FLONET is still better than TF-IDF+GPT2's S-Flo performance. This suggests that the model has acquired some general intelligence of following a flowchart, even though there is significant scope for further improvement.

Human Evaluation: We randomly sample 75 context-response pairs each from both S-Flo and U-Flo test sets and collect two sets of judgements for each pair. As we evaluate 4 systems, we collect a total of 1,200 labels from the judges. We report the human evaluation results in Table 5. We find that FLONET's relevance scores are better than the baselines for both S-Flo and U-Flo.

Knowledge Sources: To understand the contribution of each knowledge source towards response generation, we trained 3 variants of FLONET: (i) using only the dialog history (*DH*), (ii) using the dialog history and the flowchart (*DH + FC*), and (iii) using dialog history, flowchart and FAQs (*DH + FC + FAQ*). The performance is summarized in Table 6. The S-Flo trend shows both the knowledge sources contribute to the overall performance. The U-Flo numbers prove that, unsurprisingly, knowledge sources are essential for generalization to new settings, with more than 13 points increase in BLEU.

6 Analysis & Research Challenges

We now investigate FLONET errors, with the goal of identifying new research challenges posed by FLODIAL. We first manually inspect the output of the generator, *given* the retrieved document. We find that, by and large, the generator has learned its tasks well, which are deciding whether and how to use the retrieved document in generating a response. We attribute FLONET's errors primarily to the retriever. This is also apparent from Recall@1 in Table 4, which shows that FLONET makes retrieval errors for 18.6% and 33.9% of test examples in S-Flo and U-Flo, respectively.

To further diagnose retriever errors, we split

them into two categories based on whether the correct retrieval is a flowchart node or a FAQ (digression). For the former case, Table 7 reports the nature of the error. In Table 7, retrieved sibling implies the retrieved node and the correct node are sibling nodes in the flowchart. We notice that for a large fraction of errors, retriever returns a sibling node. This suggests that FLONET could not adequately ground user response to the given agent question. More surprising are 30-45% of errors that are not even in the immediate neighborhood of the true node. A much larger value for U-Flo here also suggests poor retriever generalization. Since retriever performance in this task is closely tied with the ability to follow a flowchart path, it leads to the following research question: *how can a model incorporate flowchart structure for better retriever performance?*

Table 8 analyzes retrieval errors on digressions. We find that the retriever gets a decent Recall@1 for user digressions but has a rather low performance for agent digressions. Moreover, BLEU scores suggest that generator has memorized some common digressions S-Flo, but naturally they do not generalize to U-Flo. This yields a fairly challenging research question: *how do we improve retriever performance on agent digressions?*

Finally, the challenge of zero-shot generalization to unseen flowcharts gets to the core ability of following a conversation flow, leading to the key research question: *how do we improve performance on unseen flowchart setting in* FLODIAL?

7 Conclusion

We define the novel problem of end-to-end learning of flowchart grounded task oriented dialog (TOD) for a troubleshooting scenario. We collect a new flowchart grounded TOD dataset (FLODIAL), which contains 2,738 dialogs grounded on 12 different flowcharts and 138 FAQs. We propose the first baseline solution (FLONET) for our novel task using retrieval-augmented generation. We outline novel technical challenges for TOD research identified in our work. We release FLODIAL[10] and all resources[11] for use by the research community.

Acknowledgments

Ethics Impact Statement

Crowd Worker Compensation: The crowd workers were compensated with approximately 2.5 USD for a creating paraphrases for a dialog with 15 utterances. On an average, the crowd workers spent a little less than a minute on each paraphrase. Potentially, a worker can paraphrase 4 dialogs in an hour and get compensated with 10 USD.

Intellectual Property: Flowcharts used in our data collection process are based on the flowcharts from `www.ifitjams.com`. We used these flowcharts after receiving a written permission from the creator Morris Rosenthal. We include attribution to Morris Rosenthal in the acknowledgements section.

Privacy: We now briefly describe each task used for data collection and show how the task design ensures the collected data will not contain any sensitive personal information (SPI). We would like to emphasise that the authors of the paper meticulously went over each data point collected and removed the ones that did not comply with the rules in the task description.

Problem Description Task: Each participant was provided with an artificial scenario which includes a car/laptop model, a car/laptop model year, a car/laptop related problem that they are facing. They were requested to paraphrase this information into a natural language utterance. Since the scenario was provided by us, there was almost no room for providing SPI. The paraphrases deviating from the provided details were rejected.

Paraphrasing Task: The participants were requested to create paraphrases of a given sentence. This task has no room for providing SPI.

Closing Task: The participants were asked to close a conversation between a human agent and a car/laptop user. In this task, the user and the human agent refer to each other using a second-person pronoun (e.g., I hope this was helpful to you, I am happy that this solved your problem). This task also does not involve providing any SPI.

---

[10]`https://dair-iitd.github.io/FloDial`
[11]`https://github.com/dair-iitd/FloNet`

References

Antoine Bordes and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *International Conference on Learning Representations*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 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.

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.

Layla El Asri, Hannes Schulz, Shikhar Kr Sarma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49.

Song Feng, Hui Wan, Chulaka Gunasekara, Siva Patel, Sachindra Joshi, and Luis Lastras. 2020. Doc2dial: A goal-oriented document-grounded dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8118–8128.

Revanth Gangi Reddy, Danish Contractor, Dinesh Raghu, and Sachindra Joshi. 2019. Multi-level memory for task oriented dialogs. 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 3744–3754, Minneapolis, Minnesota. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Z. Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

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.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Word-based dialog state tracking with re- current neural networks. In *In Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

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.

Youxuan Jiang, Jonathan K. Kummerfeld, and Walter S. Lasecki. 2017. Understanding task design trade-offs in crowdsourced paraphrase collection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 103–109, Vancouver, Canada. Association for Computational Linguistics.

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.

Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu, and Dilek Hakkani-Tur. 2020. Beyond domain apis: Task-oriented conversational modeling with unstructured knowledge access. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 278–289.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

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, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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

Dinesh Raghu, Nikhil Gupta, and Mausam. 2019. Disentangling Language and Knowledge in Task-Oriented Dialogs. 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 1239–1255, Minneapolis, Minnesota. Association for Computational Linguistics.

Dinesh Raghu, Atishya Jain, Mausam, and Sachindra Joshi. 2021. Constraint based knowledge base distillation in end-to-end task oriented dialogs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5051–5061, Online. Association for Computational Linguistics.

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

Deborah Rossen-Knill, Beverly Spejewski, Beth Ann Hockey, Stephen Isard, and Matthew Stone. 1997. Yes/no questions and answers in the map task corpus. Technical report, University of Pennsylvania, Institute for Research in Cognitive Science.

Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. *CoRR*, abs/1507.02221.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *Proceedings of the International Conference on Machine Learning, Deep Learning Workshop*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Lina M. Rojas Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young. 2016. Conditional generation and snapshot learning in neural dialogue systems. In *EMNLP*, pages 2153–2162, Austin, Texas. ACL.

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

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. 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.

Mohammad-Ali Yaghoub-Zadeh-Fard, Boualem Benatallah, Moshe Chai Barukh, and Shayan Zamanirad. 2019. A study of incorrect paraphrases in crowd-sourced user utterances. 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 295–306, Minneapolis, Minnesota. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational*

Linguistics: System Demonstrations, pages 270–278.

Tiancheng Zhao and Maxine Eskenazi. 2018. Zero-shot dialog generation with cross-domain latent actions. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 1–10.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390, Online. Association for Computational Linguistics.

A Appendix

A.1 Training Details

FLONET is trained in two phases: pre-train and fine-tune. In the pre-train phase, we use recall@1 and BLEU as the early stop criteria for the retriever and generator respectively. The recall is computed using the weakly supervised labels. The hyper-parameters (embedding size, lr_R, lr_G, dropout) that achieved the best validation numbers in pre-train stage were (100, 1E-4, 6.25E-5, 0.01) and (200, 1E-4, 6.25E-5, 0) for S-Flo and U-Flo respectively. The best hyper-parameters that achieved the best BLEU in fine tune phase were (100, 1E-4, 2.5E-6, 0) and (200, 1E-5, 2.5E-7, 0) for S-Flo and U-Flo respectively. The BLEU scores on held-out validation set were 16.62 and 9.72 for S-Flo and U-Flo respectively. We use AdamW optimizer (Kingma and Ba, 2014) for training and beam search for decoding with beam width of 5 and a maximum decode length set to 60 tokens.

All experiments were run on a single Nvidia V100 GPU with 32GB of memory. The S-Flo retriever, U-Flo retriever and generator have 3M, 23M and 117M trainable parameters respectively. Thus, FLONET has a total of 120M trainable parameters for S-Flo and 140M for U-Flo. FLONET has an average runtime of approximately 7 hours (80 mins per epoch) and 8 hours (82 mins per epoch) for S-Flo and U-Flo respectively.

A.2 GPT2 Input

FLONET uses GPT2 as the generator. Following Wolf et al. (2019), our input is constructed as shown in Figure 3. During inference, GPT2 take the retrieved document concatenated with the sequence of utterances from the dialog history as the input and predicts the agent response.

Component Type	Constructed Component
\<Retrieved Document\>	[BOS] Is there a pinhole leak in hose or at clamps?
\<User Utterance 1\>	[S1] I am facing a problem with my car's engine. It seems to be overheating. My car is a Ford which I have had for 3 years. Can you please help me?
\<Agent Utterance 1\>	[S2] Is there any steam coming from your car's engine?
...	...
\<Last User Utterance\>	[S1] I don't see any leak in my car's radiator.
\<Agent Response\>	[S2] Does the hose or clamps have any leak?

Figure 3: Breakdown of the components in GPT2 input and output.

A.3 GPT2 Inference

We experiment with various inference settings and used the setting which performed the best in the validation set. We tried two decoding techniques: nucleus sampling (Holtzman et al., 2020) with top-p as 0.9 and beam search with beam width of 5. We also experimented with the number of top-k documents to be used. In the case of Top-1 decoding, we use only the top retrieved document to generate response candidates. For Top-5, we take the top 5 retrieved documents and generate candidate responses from each document. Each candidate response score is computed as a product of the probability of generating the candidate given the retrieved document $\prod_{t=1}^{T} p_\theta(y_t|\mathbf{h}, z, y_{1:t-1})$ and the probability of the retrieved document $p_\eta(z|\mathbf{h})$. Lastly, we experimented with response length normalization to avoid favouring shorter sequences. The probability of each candidate is given by $(\prod_{t=1}^{T} p_\theta(y_t|\mathbf{h}, z, y_{1:t-1}))^{1/T}$ where T is the length of the candidate. The validation and test BLEU scores of various settings on the S-Flo split is shown in table 9. We see that beam search on top-1 document with length normalization resulted in the best validation BLEU.

Decoding Technique	Top-k	Length Norm.	BLEU Val	BLEU Test
Nucleus	Top-5	No	10.77	13.41
		Yes	14.13	14.21
	Top-1	N/A	16.62	16.34
Beam	Top-5	No	17.87	17.42
		Yes	18.44	17.19
	Top-1	No	18.94	18.35
		Yes	**20.41**	**19.46**

Table 9: Validation and test BLEU scores of various settings on the S-Flo split.

A.4 Qualitative Examples

Table 11 and 10 shows responses generated by various systems on examples from U-Flo and S-Flo test set respectively. In Table 10, we see that FLONET and FLONET (No PT) generates responses similar to the gold response as they were able to generalize to unseen flowcharts.

A.5 Example Dialogs from FLODIAL

Three randomly selected dialogs from FLODIAL are shown in Table 12. The first dialog is grounded on the *wireless network troubleshooting* flowchart, sec-

ond is grounded on *steering problems* flowchart and the last is is grounded on *car won't start* flowchart.

A.6 AMT Tasks

Figure 4, 5, 6 and 7 show the instructions and examples provided to crowd-workers for problem description, non-contextual paraphrasing, contextual paraphrasing and closing task respectively.

A.7 FAQ Collection

For each flowchart we collected a set of FAQs associated with it. We collect the FAQs in two steps: question collection and answer collection. In the question collection step, we displayed each flowchart node question to novice (laptop or car) users and asked them if they understand the node question. If not, they were requested to come up with a follow-up question that would help them better understand the node question. We aggregated all the follow-up questions to create the set of FAQs associated with that flowchart. In the answer collection step, in-house (laptop or car) expert provided answers for the questions collected in the first step.

User	I've had my Infiniti QX60 for 6 years now. It's recently been making awful noises. I got stopped by the police on my way to work because the noise was no loud. Please help me fix this.
Agent	The sound that is coming from your car, is it a clunking sound?
User	I do hear a clunking sound coming from the car.
Gold Response	Is it only when you drive over a bump or a dip that the car makes a noise?
TF-IDF + GPT2	Are the ticking noises heard only when the car is moving?
FLONET (No PT)	Does the car make a noise only when you drive over bumps or dips on the road?
FLONET	Does the car make the noise only when you drive it over a bump or a dip on the road?

Table 10: Responses predicted by various algorithms for an example from the S-Flo test set. The highlighted text shows that TF-IDF + GPT2 missed out on an important detail.

User	My Dell G5 15 5590 is having some issues. I got it about a year ago but within the last month the battery just refuses to work properly. I don't understand what is happening here. This happens even if I plug it in overnight.
Agent	When the charger isn't plugged in, can the laptop power up?
User	I can see that it can start up in that case.
Agent	Do you find that the battery in your laptop quickly drains of power?
User	The battery drains too fast.
Agent	There will be a manufacture date on the laptop's battery. Please check if it is older than a year and let me know.
User	How do I check that date on my battery?
Gold Response	In order to locate the manufacturing date, you need to remove the battery and find a sticker on it with a multi-field part number. The fourth field in the number identifies the date.
TF-IDF + GPT2	You can use a multimeter to measure the voltage of the AC power adapter. The voltage should be the same as the voltage printed on the adapter.I am glad I could help. Have
FLONET (No PT)	Remove the battery and you will find a sticker on it with a multi-field part number.
FLONET	Please remove the battery and you will find a sticker on it with a multi-field part number. The fourth field in the part number corresponds to the manufacturing date.

Table 11: Responses predicted by various algorithms for an example from the U-Flo test set. The highlighted text points out that FLONET generated the best response compared to other models.

User	Due to the pandemic, I am working from home right now. I have not been able to get my work done for the last few days. I am falling behind and missed an important deadline. My computer will not let me access the internet. I have an Acer Swift 3 that is two years old. Can you help me?
Agent	Does your laptop show your wireless network in the list of available networks?
User	I can see the wireless network on my laptop.
Agent	Do you know the method to disable the router's security settings?
User	No, I do not know it.
Agent	Are you able to access the router setttings page?
User	I am unable to access the router settings page.
Agent	Do you remember the router's login information?
User	I do remember the login information.
Agent	From a device that is connected to the network, type 192.168.1.1 or 192.168.0.1 in the browser. Once you do that, you will be able to see the router setting page where you can login and disable the security settings. Are you able to connect to the network now?
User	I did what you said. I still can't connect to the network.
Agent	Can you access the internet if you link your laptop to the router with an ethernet cable?
User	Yes, doing that establishes Internet access.
Agent	Do you see default settings for your router right now?
User	I can see the default settings.
Agent	Do you have an intermittent connection?
User	It is not intermittent. It is very steady.
Agent	Do you how how to check what networking protocols are being used by your laptop?
User	I don't know how. I have never tried to do that before.
Agent	Go to 'Network and Internet' in your Settings menu. Then go to 'Advanced Options' in WiFi settings and check the protocol that is in use. It should be present in the description field. Please tell me if it is 802.11g and the same as the router.
User	It is the exact same as the router.
Agent	Does your laptop allow you to connect to the internet using public Wi-Fi networks?
User	It's different when connecting to a public Wi-Fi network. The laptop connects then.
Agent	The wireless router you are using is malfunctioning, and needs to be replaced.
User	That's a shame. I guess I'll go out and get a new router in the morning. Thanks for letting me know.
Agent	I was happy to help. Take care and best of luck!

User	my Honda Accord has some steering problems. I have had the car for almost 3 years now. I hope you are able to diagnose this so that I can fix my car.
Agent	Please turn the steering wheel a few inches. Are the tires not moving?
User	You are right, the tires aren't moving when I turn the steering.
Agent	Do any of the wheels have lug nuts that aren't tightly secured?
User	I just checked the wheels. None have lug nuts that aren't tightly secured.
Agent	Do you see a loose axle nut on any of the wheels?
User	I don't know what an axle nut is or where it is located.
Agent	The axle nut keeps the wheel on the axle. You can find it under the hub cover at the center of the wheel.
User	I can't find a loose axle nut.
Agent	Does your car feel abnormally bumpy if you go over potholes or an uneven road?
User	yes, the car feels abnormally bumpy.
Agent	You should get the strut of your suspension replaced because it has failed.
User	Thanks, replacing the strut resolved the problem.
Agent	You're welcome and have a great day!

User	I can't start my car and it doesn't respond when I turn the key. It's come to my attention that when turning the key, the started doesn't spin. The make/model of my car is a BMW X7. It recently gave me a lot of trouble because I got stranded at work. Even worse, it was on a Friday, so I couldn't go home to start my weekend on time. I finished my work and clocked out. I walked out to my car, excited to go home and start my weekend. I got to my car in the parking lot and turned the key. When I did, the car did not respond. There must be some kind of problem with the starter because when I turn the key in the ignition, the starter doesn't spin.
Agent	Using a voltmeter, do you know how to measure your car battery voltage?
User	Yes, I can do that using a voltmeter.
Agent	Turn the headlights on and please tell me if the battery measures more than 12V.
User	The battery is giving more than 12V after turning on the headlights.
Agent	Did you check the battery terminals to see if they are unclean?
User	Yes I checked the battery terminals and they are unclean.
Agent	Your battery terminals need to be cleaned, so clean the terminals, connectors and engine ground.
User	Yeah, I cleaned them out, thanks for the help!
Agent	No problem at all, have a good day!

Table 12: Sample dialogs from FLODIAL

Task 1 - Problem Description

Each HIT will contain a primary issue with your car. Please describe the issue as completely as possible.

Along with the primary issues, you may also be provided with (i) a secondary issue, (ii) car details (model, manufacturer) and (ii) additional requirements (like fabricate a scenario about the trouble you had to go through because of this issue).

Dos:

1. The secondary issue, car details and additional requirements should be used in the description when provided
2. The issue should be described with atleast two sentences
3. Please feel free to be creative when writing the description

Donts:

1. Do not copy-paste a description from the web. Such descriptions will be REJECTED.
2. Do not include problems other than the primary and secondary issue.
3. Description should contain the issues provided. Descriptions about other random issues will be REJECTED.
4. Do not use the same sentence structure as given in the summary. Re-phrase them using synonyms, change the verb tense etc.
5. The description should not be in the same order of the summary. For example, dont state the issue first, then the car details and finally the additional requirements. Mix them up.

Good examples	**Bad examples**

Summary

Primary Issue: The car's brake pad is worn out.

Secondary Issue: Everytime you brake there is a screeching sound.

Car Details: You own a Cadillac XTS and you have had it for the last 2 years.

Additional Requirement: Fabricate a scenario about the trouble you had to go through because of this issue.

Description

My Cadillac XTS is making a screeching sound everytime I press the brake. I think the brake pads of my car are worn out. I have had this car for 2 years but have never faced this issue before. Driving home from work with this screeching sound was so embarrassing.

Summary

Primary Issue: The car's brake pad is worn out.

Secondary Issue: Everytime you brake there is a screeching sound.

Car Details: You own a Cadillac XTS and you have had it for the last 2 years.

Additional Requirement: Fabricate a scenario about the trouble you had to go through because of this issue.

Description

Car's brake pad is worn out. I also noticed that everytime I brake there is a screeching sound. I own a Cadillac XTS and I have had it for the last 2 years. I had to go through trouble because of this issue.

Figure 4: Instructions provided to AMT workers for the problem description task.

Task 2 - Non-Contextual Paraphrasing

Each HIT will contain a question about a car. Please rephrase the given question in two different ways. Please refer to the tips for paraphrasing section below.

Dos:

1. Make sure you always phrase as a question.
2. Use a polite and formal tone while rephrasing.
3. The paraphrases should be grammatical correct.

Donts:

1. Do not repeat the same given sentence or use the same rephrasing twice. Tasks with repetition will be rejected.
2. Paraphrases with one or two words changed in the given sentence will be rejected as well.

Tips to Paraphrasing:

1. *Change the grammatical structure*
    - Sentence: The metal beams were corroded by the saltwater.
    - Paraphrase: The saltwater corroded the metal beams.
2. *Change the form of words. In this example verb has been replaced by noun*
    - Sentence: The metal beams were corroded by the saltwater.
    - Paraphrase: The corrosion on the metal beam was caused due to the saltwater.
3. *Use synonyms for words/phrases*
    - Sentence: The metal beams were corroded by the saltwater.
    - Paraphrase: The seawater has rusted the metal beams.

Good examples	Bad examples
Question: Do the tires make a screeching sound when you brake?	**Question**: Do the tires make a screeching sound when you brake?
Paraphrase 1: Do you hear a screeching sound from the tires when you brake?	**Paraphrase 1**: Do the tires make a screeching sound when you brake?
Paraphrase 2: When you apply brakes, is there a screeching sound from the tires?	**Paraphrase 2**: Do the tires make a screeching sound when you apply brake?

Figure 5: Instructions provided to AMT workers for the non-contextual paraphrasing task.

Task 3 - Contextual Paraphrasing

You will be given 3 yes/no questions and the polarity of each answer. You have to phrase a response for each question. The response should follow a given set of rules. Questions are asked by a customer support agent, and you have to respond as a customer.

Dos:

1. The response has to be a complete sentence.
2. The response should have the mentioned polarity.

Donts:

1. Simple 'Yes' and 'No' phrases will be rejected.
2. Responses that don't follow the specified rules will be rejected.
3. Even if the question, polarity and rules are repeated, the response should be different.

Good examples	Bad examples
Support Agent Q1: Did you replace your brake shoes less than a year ago?	**Support Agent Q1**: Did you replace your brake shoes less than a year ago?
Response Polarity: Yes	**Response Polarity**: Yes
Rules:	**Rules**:
• Don't use pronouns to refer things mentioned in the question	• Don't use pronouns to refer things mentioned in the question
Customer Response 1: I remember replacing the brake shoes about 6 months ago.	**Customer Response 1**: Yes.
Support Agent Q2: Do you hear flapping noise from your engine?	**Support Agent Q2**: Do you hear flapping noise from your engine?
Response Polarity: No	**Response Polarity**: Yes
Rules:	**Rules**:
• Don't include the words Yes/No in your response. • Use pronouns to refer things mentioned in the question • Describe it using two sentences	• Use pronouns to refer things mentioned in the question • Describe it using two sentences
Customer Response 2: I just turned it on and checked. It isn't making any such noise.	**Customer Response 2**: Yes, it does.

Figure 6: Instructions provided to AMT workers for the contextual paraphrasing task.

Task 4 - Closing

Each HIT will contain a solution from a customer support agent. First, you will act as a customer and respond to the solution. Then, you will act as a support agent and respond the the customer's utterance that you wrote. Make sure the agent's response ends the conversation. You will be given rules for each utterance.

Dos:

1. You have to follow the given rules for each utterance.
2. Use a formal and warm tone for the support agent.
3. Make sure the conversation ends.
4. Make sure the conversation has a natural flow and is coherent.
5. The utterances should be elaborate and specific to the given agent solution.

Donts:

1. Do not copy-paste the given rules or a random sentence from the internet. Such tasks will be REJECTED.
2. Tasks which do not follow the rules will be REJECTED
3. Utterances with just one or two words will be REJECTED.

Good examples	**Bad examples**
Support Agent: The issue is with your car's headlight. Please replace it to fix your problem.	**Support Agent**: The issue is with your car's headlight. Please replace it to fix your problem.
Customer:	**Customer:**
Rule: Phrase customer's response to the above agent utterance. The response should (1) mention that the agent's solution worked, and (2) thank the agent for the help.	*Rule: Phrase customer's response to the above agent utterance. The response should (1) mention that the agent's solution worked, and (2) thank the agent for the help.*
Oh I see the headlight was the culprit. I will be able to fix it. Thanks a lot for your help.	*use it to phrase a client response thanking the agent for help and indicate it worked.*
Support Agent:	**Support Agent:**
Rule: Phrase support agent's response. (1) Respond to the customer's thank you. (2) End the conversation with a warm note.	*Rule: Phrase support agent's response. (1) Respond to the customer's thank you. (2) End the conversation with a warm note.*
I am glad that worked out. It was my pleasure to help you out. Have a good day. Goodbye.	*Glad. Thanks.*

Figure 7: Instructions provided to AMT workers for the closing task.