Abstract

Targeted-guided response generation enables dialogue systems to smoothly guide a conversation from a dialogue context toward a target sentence. Such control is useful for designing dialogue systems that direct a conversation toward specific goals, e.g., such as providing counselling and creating non-obtrusive recommendations. In this paper, we introduce a new technique for target-guided response generation, which first finds a bridging path of commonsense knowledge concepts between the source and target, and then uses the identified bridging path to generate transition responses. Additionally, we propose techniques to re-purpose existing dialog datasets for target-guided generation. Finally, we demonstrate the shortcomings of existing automated metrics for this task, and propose a novel evaluation metric that we show is more effective for target-guided response evaluation. Our experiments show that our proposed evaluation metric is reliable and our techniques outperform baselines on the generation task. Our work generally enables dialogue system designers to exercise more control over the conversations that their systems produce.

1 Introduction

Open-domain conversational systems have made significant progress in generating good quality responses driven by strong pre-trained language models (Radford et al., 2019; Devlin et al., 2019) and large-scale corpora available for training such models. However, instead of passively responding to a user, many practical dialogue system applications operating in domains such as conversational recommendation, hospitality and education have specific goals to achieve. Prior work have used mechanisms such as emotion labels (Zhong et al., 2019), persona (Song et al., 2019), and politeness (Niu and Bansal, 2018) to control the conversations towards system agenda. However, such approaches require labelled training data for a fixed set of coarse-level labels, making it harder to incorporate new goals in a system. In this work, we study the problem of proactive response generation based on a target sentence or instruction. For example in Figure 1, given the context ‘I enjoy swimming’, the system guides the conversation towards the target ‘I like to travel to new places’ by mentioning ‘I like to swim at beaches when I go on vacation’. Using target sentences for proactive control is a intuitive and flexible control mechanism for dialogue developers, free of domain-specific handcrafting and annotations.

Existing publicly available dialogue corpora generally consists of free-flow conversations where the speakers move the conversation forward based on the dialogue history instead of an agenda. We build upon the recently released Otters dataset (Sevegnani et al., 2021) with one-turn topic transitions for mixed-initiative in open-domain conversations. Given a source sentence from a speaker, the task is to generate a topic transition sentence with “bridging” strategies to a target sentence from another speaker. The task is challenging on several fronts. Firstly, the system needs to balance the trade-off be-
tween coherence with the context while smoothly transitioning towards the target. Secondly, the Otters training dataset is relatively small (less than 2000 training instances), making it a low-resource setting. Thirdly, there are no good established automated metrics for this task, as the standard word-overlap metrics are insufficient in this task.

In this work, we propose methods to leverage commonsense knowledge from ConceptNet (Speer et al., 2017a) to improve the quality of transition response. Our technique decomposes the response generation process into first generating explicit commonsense paths between the source and target concepts, followed by conditioning on the generated paths for the response generation. This is intended to mimic how humans might bridge concepts for creating transitions in conversations using commonsense knowledge. This technique offers two benefits: 1) Leveraging external ConceptNet knowledge solves the data scarcity issue and improves the reasoning strategies, leading to fewer illogical transitions; 2) Since the transition response is grounded on commonsense knowledge paths, the explicit paths used by the model can provide explanations for the concepts used by the model, as well as provide control over the generation process. Furthermore, we propose a data augmentation mechanism to help with the data scarcity issue by re-purposing training data from DailyDialog, an open-domain dialogue dataset. Both these approaches are complementary and outperform existing baselines in response quality and transition smoothness. We demonstrate how the proposed approach of using explicit bridging paths enables improved quality of transitions through qualitative and human studies.

Automated evaluation is a challenging aspect in dialogue response generation tasks (Zhao et al., 2017). We show that the existing word-overlap metrics such as BLEU can be easily fooled to give high scores for poor quality outputs in this task. We propose a metric TARGET-COHERENCE which is trained using hard adversarial negative instances, and achieves high correlation with human judgement ratings of system outputs. As part of this work, we collect and release a dataset of human ratings of various system outputs for this task.

2 Related Work

Target Guided Dialogue Response Generation: Sevegnani et al. (2021) is perhaps the closest to our work described in this paper. They work on the task of generating a new utterance which can achieve a smooth transition between the previous turn’s topic and the given target topic. Past work in controllable text generation has explored steering neural text generation model outputs to contain a specific keyword (Keskar et al., 2019), a graph (Wu et al., 2019), or a topic (Ling et al., 2021). Steering dialogue towards a given keyword has also been explored in past work (Tang et al., 2019; Qin et al., 2020a; Zhong et al., 2021), albeit as a retrieval task. Compared to these, our goal is to generate a next utterance in a dialogue setup which can steer a conversation towards target sentence in a smooth fashion rather than generating an utterance belonging to a given topic. Our work is also related to prior work on text infilling (Donahue et al., 2020; Qin et al., 2020b), though compared to them we work in a dialogue setup and utilize commonsense knowledge to perform the infilling.

Commonsense for Dialogue Generation: Commonsense knowledge resources (Speer et al., 2017b; Malaviya et al., 2020) have been used successfully in dialogue response generation for tasks such as persona-grounded dialogue (Majumder et al., 2020) and open-domain topical dialogue generation (Ghazvininejad et al., 2018). Zhou et al. (2021) created a dataset focusing on social commonsense inferences in dialogue and Arabshahi et al. (2020) design a theorem prover for if-then-because reasoning in conversations. More broadly, commonsense knowledge has been used in other text generation tasks such as story-ending and essay generation (Guan et al., 2019a; Yang et al., 2019).

Automated Metrics for Evaluating Dialogue Quality: Automated metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and BertScore (Zhang et al., 2020) are widely used to evaluate quality of machine-generated text. However, such metrics often correlate poorly with human judgement ratings of generated text quality (Sai et al., 2020). Past work has explored trained model-based metrics such as ADEM (Lowe et al., 2017) and RUBER (Tao et al., 2017). However, training such model-based metrics often relies on tagged training data. Gupta et al. (2021) propose ways to mitigate the need for such labelled data by automatically synthesizing negative examples. Our proposed metric is along similar lines, though we utilize different techniques for synthetic negative example generation.
3 Task Overview

We first formalize the task of target-guided response generation. Given a conversation history of n utterances $C = \{u_1, u_2, ..., u_n\}$ between two speakers A and B, and a target t for speaker B’s turn $u_n$, the task is to generate a transition sentence $s_c$ which serves as a smooth link between the context and the target. The target can be defined in terms of a phrase or a sentence. Opper dataset (Sevegnani et al., 2021) consists of a simplified setting of one-turn topic transitions, where the conversation history consists of a single utterance $u_a$, and a target utterance $u_b$ and the task is to generate a transition utterance $s$ to serve as a smooth link between $u_a$ and $u_b$. The task is challenging since a system needs to devise a strategy that balances the competitive objectives of generating a response which acknowledges and is coherent to the context, while smoothly driving the conversation towards the target.

In this work, we propose two approaches for the transition response generation task: 1) Commonsense-guided response generation (section 4), and 2) Data augmentation to tackle data sparsity (section 5). We refer to the proposed method as CODA (Commonsense Path and Data Augmentation). Furthermore, we propose a novel metric TARGET-COHERENCE to automatically evaluate the smoothness of response transitions (section 6).

4 Commonsense-Guided Response Generation

We frame the target-guided response generation task as follows. Given a conversation history of n utterances $C = \{u_1, u_2, ..., u_n\}$ and a target $t$, a conditional language model learns to predict the tokens of the transition response $s$ by minimizing the cross entropy loss of the ground truth transition response.

As mentioned previously, target-guided generation can potentially benefit by incorporating commonsense reasoning. Pre-trained models are known to suffer in cases where commonsense knowledge is required during generation (Zhou et al., 2018; Guan et al., 2019b), especially in tasks where there is not enough data available for learning commonsense patterns from the text, which is true for our case. In contrast, Commonsense Knowledge Graphs like ConceptNet (Speer et al., 2017a) provide structured knowledge about entities, which enables higher-level reasoning about concepts. In this work we use commonsense knowledge from ConceptNet for planning a transition response. ConceptNet is a large-scale semantic graph that has general phrases as nodes and the commonsense relationships between them, such as ‘IsA’ and ‘At-Location’. However, ConceptNet consists of non-canonicalized text and hence suffers from severe sparsity (Malaviya et al., 2020). Therefore, it is not always possible to find the concepts and connections between context and target concepts.

To address the sparsity issue, we develop Knowledge Path Generator (KPG), a language model that generates knowledge instead of retrieving it from KG. The model takes a pair of entities or concepts as input and generates a multi-hop path connecting the two. Since the knowledge is generated, the path may not exist in ConceptNet and may contain nodes not actually present in KG. Thus the generated knowledge generalizes over the facts stored in the KG (Details in Section 4.1).

To generate commonsense based responses, we train a Commonsense Response Generator (CRG) model to generate the transition response conditioned on the paths generated by the KPG model. Conditioning the response generation on commonsense paths improves the reasoning capabilities of the CRG model and provides the added benefits of interpretability and control over the generation process. Figure 2 represents the overview of our proposed approach.

4.1 Commonsense path generator

The objective of the KPG model is to connect an entity phrase or topic from the context with an entity from the target by creating knowledge paths between them.

Path Sampling: To create training data for the KPG model, we sample paths between entity phrases from ConceptNet using random walks. This step builds upon past work of Wang et al. (2020). Given the ConceptNet graph with a set of nodes $N$ and edges $E$, we perform random walks on the graph to sample a set of paths $P$ of the form $p = \{n_0, e_0, n_1, e_1, ..., e_{k-1}, n_k\} \in P$. Here, a path $p$ connects a head entity phrase $n_0$ with the tail entity phrase $n_k$ via intermediate entities and edges (or relations) $e_i$. To sample paths, the random walk begins with a random entity node $n_0$ and samples a path of random length $k$ in the set $K = 6$, where we have set $k = 6$ in this work.
When using CRG at inference time, we use a language model (referred to as KPG-h) to construct commonsense paths linking a head and a tail entity. For the KPG-h model, the input is just the head entity \( h \) and the tail entity \( t \). For a sample path \( p = \{h, e_0, n_1, e_1, \ldots, c_{k-1}, n_t\} \), the path is formatted into the sequence “[wc] k1 [wc] k2 … [target] n_t [sep] n_h e_0 n_1 e_1, \ldots, c_{k-1} n_t”. Here “wc” symbolizes “will contain”. The set \( E_p = \{k_1, k_2, \ldots, k_n\} \) used in the sequence is a randomly permuted sequence of entities \( n_1, n_2, \ldots, n_{k-1} \) of the sampled path. Training with this sequence indicates to the model that the path generated between \( n_h \) and \( n_t \) should contain the entities from the set \( E_p \) in a sensible order. Specifying the special token “[target]” followed by the tail entity \( n_t \) informs the model about the last entity it should output when generating a path. We discuss how the set \( E_p \) is constructed for training the CRG model in the next section.

### 4.2 Response generator

The Commonsense response generator samples and uses it for generating commonsense knowledge conditioned transition responses.

#### Entity extraction

We extract a set of entities \( E_h, E_t \) and \( E_r \) from the context, target and gold transition response respectively. We first run NLTK’s part-of-speech tagger on a sentence, and then use NLTK’s chunker to extract the set of noun and verb phrases present in the sentence. Additionally, we design simple grammar rules (details in Appendix) to convert some phrases to more concise forms (for example, “watching the star” is converted to “watch stars”). This step is done to make the entities more similar to the kind of nodes present in ConceptNet.

#### Sampling and filtering paths: In this step, for

[Figure 2: Overview of Commonsense Response Generator (CRG) model: During training, the Knowledge Path Generator model KPG-wc is fed the entities from the context, target and the gold transition response, and the output path from KPG-wc is used in CRG model’s training. During inference, KPG-h model is fed the context and target entities and the model generates a path with new entities such as “vacation”. CRG model conditions on this path for transition response generation.]
every pair of head and tail entity from \( E_h \) and \( E_t \), we sample multiple paths from the KGP models using topk sampling and chose one or more of these paths for training and inference. For training the CRG models with the commonsense paths, we need tocurate paths that are relevant to and aligned with the gold response so that they are not ignored by the CRG model during inference. We achieve this by first sampling paths which are relevant to the ground truth response, and then apply filtering mechanisms to curate the final set of paths. For training data path sampling, we use the \textit{KPG-wc} model. The input to the model is a head and tail entity pair \( n_h \) and \( n_t \), and the entity set \( E_p \) that consists of the set of entities \( E_r \) from the gold transition response. The model then generates a set of paths that contain the head and tail entities as well as the gold response keywords. Thus, the sampled path is inherently relevant to the gold response due to the conditioning on gold keyword entities. During inference, the set \( E_r \) is not available, so we leverage the \textit{KPG-ht} model that takes just the head and tail entity pair \( n_h \) and \( n_t \) as input to generate a commonsense path.

Assuming the context and target consists of \( m \) and \( n \) entities each, and we sample \( q \) number of paths per pair, we get a total of \( m \times n \times q \) number of paths for each data instance. Since \( m \times n \times q \) can be a large number, we use simple methods to sub-select entity pairs and paths. (1) Sub-selecting Entity Pairs: We score an entity pair by calculating the inverse document frequencies (computed using Gutenberg English corpus) of the entity tokens and summing up the maximum value found for a token in each entity in the pair. For training phase, we keep the top \( D \) pairs of entities, and for testing phase we keep only the highest-scoring pair. (2) Sub-selecting paths: We apply the following strategies to prune the set of paths for each entity pair: 1) \textit{Perplexity} - We filter out all the paths whose perplexity values (form the KGP models) are more than double the average perplexity values of all paths between an entity pair. 2) We remove all the paths which have repetition of entities. 3) For paths in training data, we filter out paths which contain entities not present in the gold response. After filtering out such paths, we have a final set of \( P \) paths per response. The paths from set \( P \) are converted into natural language by converting the relation and inverse relations into textual format. For example, “art gallery UsedFor for art” is converted to “art gallery is used for art”.

**Training and inference in CRG model.** The CRG model is trained as a conditional model with the following input sequence: “knowledge path [target] target sentence [context] context sentence [response] transition response” for each knowledge path from the set \( P \). We train the CRG model by minimizing the log-likelihood loss of the transition response \( r \) given the context \( C \), target \( T \), and the path \( p \). For inference, we first create the set of paths \( P \) by entity extraction, path sampling and filtering and choose a random path \( p \) from the final set \( P \). The model then generates the transition response conditioned on the sequence of \( c, t, \) and \( p \).

## 5 Data Augmentation

The task of target sentence guided response generation is still a relatively unexplored task, and Otters (Sevegnani et al., 2021) is the only suitable dataset for this task to the best of our knowledge. However, Otters is small and consists of only a few hundred context-target pairs with a few transition responses for each pair. This makes learning transition concepts and strategies challenging in this low-resource setup. On the other hand, there are many publicly available dialogue datasets for training response generation models. Such datasets contain free-flow conversations, where although the speakers generate context coherent responses, but they do not condition their responses on any target. We propose a technique to leverage and re-purpose such datasets for the task of target-guided dialogue generation. We pick the Dailydialog (Li et al., 2017) dataset for experimentation and convert its conversations to target-guided conversations in two steps: 1) Target creation, and 2) Data filtering.

| CONTEXT \( c \) | Is my booking complete? |
| RESPONSE \( r \) | Your reservation is confirmed. Now I need your phone number |
| SRL output example | agent=I predicate=need instrument=your number |
| TARGET clause \( t \) | I need your phone number |

Figure 3: An example to demonstrate how a conversation in DailyDialog can be re-purposed for the task of target-guided response generation.

For target creation, given a dialogue context \( c \) and its response \( r \), we first break the response \( r \) into sentence clauses. An example target creation is shown in Figure 3, showing how we break a response into sentence clauses, and pick one of the
clauses as target. (Details about clause identification in Appendix A.1) For each predicate identified in a sentence, we create a clause by putting together the predicate and arguments in a textual sequence. Finally, we only use the clause occurring towards the end of the response as a target.

The target creation step does not guarantee that a candidate response transitions smoothly towards the target clause. In the example above, the transition response "your reservation is confirmed." is coherent to the context, but does not transition well towards the target. In data filtering step, we use a TARGET-COHERENCE metric to score a transition response \( r \) in terms of its coherence to the context \( c \) and the smoothness towards the target \( t \). The metric is described in more detail in section 6. The metric assigns a score between 0-1 for a transition response and we remove instances with a score less than a threshold \( k \) (set to 0.7) from consideration.

The remaining instances are used for pretraining response generation models which are finally fine-tuned on the Otters dataset.

6 Target-Coherence Metric

Evaluating target-guided responses is a challenging task as a good transition response needs to be both - coherent to the context and smoothly transitions towards the target. Furthermore, since the task is open-domain and open-ended, there are many possible correct responses which may not match with a reference response \( (\text{Çelikyilmaz et al., 2020}) \). To tackle both these challenges, we propose a machine-learned model for this task that does not use human written references for evaluation. The proposed metric named TARGET-COHERENCE is based on a classification model that is trained to classify a transition response as either positive, that is, it is coherent to the context and smoothly transitions towards the target, or negative, that is, the response is either not coherent to the context or is not able to transition towards the target.

We use the gold transition response from the training dataset to create positive instances for training. For a positive instance with context \( c \), target \( t \) and response \( r \), we create negative instances using the following mechanisms: 1) We hold two out of \((c, t, r)\) constant while randomly sample the third one. For example, sample a random context \( c' \), which makes \( r \) incoherent to the \( c' \). 2) We use a GPT-2 model trained on Otters dataset to generate a response \( r' \) coherent to \( c \) but conditioned on a random target \( t' \). 3) For a given target \( t \), we chose a response \( r' \) from the Otters training set which has \( t \) as the target but context \( c' \neq c \). We sample a maximum of 2 negative instance per mechanism and balance the count of positive and negative instances by repeating positive instances.

We fine-tune a pre-trained BERT-base \( (\text{Devlin et al., 2019}) \) model using this set of instances with binary cross entropy loss.

7 Experiments

7.1 Datasets

We use two datasets in our experiments. 1) Otters \( (\text{Sevegnani et al., 2021}) \) contains instances with context-target-transition response triplets. It consists of two sets of splits. The Out-Of-Domain (OOD) split ensures that none of the context-target pairs in the test set are present in the train set. In the In-Domain (ID) split, one of either the context or the target in each pair in the test-set is allowed to appear in the train-set. Otters dataset consists of multiple responses per context-target pair. Dailydialog dataset consists of casual conversations between two speakers. In Table 1 we present the number of dialogues for dailydialog dataset and number of responses for otters with number of unique context-target pairs in brackets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Otters-id</td>
<td>1,929 (693)</td>
<td>1,160 (404)</td>
<td>1,158 (303)</td>
</tr>
<tr>
<td>Otters-ood</td>
<td>2,034 (677)</td>
<td>1,152 (372)</td>
<td>1,130 (372)</td>
</tr>
<tr>
<td>Dailydialog</td>
<td>11,118</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table 1: Overview of the datasets.

7.2 Baselines for generation

We report results for the proposed model CODA and several of its variants:

- **CODA-NoCSKB**: Variant of CODA without the use of explicit commonsense paths.
- **CODA-NODA**: Variant of CODA trained without additional data from DailyDialog.
- **CODA-KEYWORDS**: Variant of CODA that ignores the edge types (such as ‘at location’) in the knowledge path.
- **CODA-Upper**: Variant of CODA which uses the path inferred from the gold response using the KPG-wc keywords model during inference. It establishes a upper-bound for the CODA model.

We report results for a number of baselines:

- **GPT-2**: \( (\text{Radford et al., 2019}) \) A pretrained GPT-2-small language model fine-tuned on Otters data.
We report standard automated metrics such as BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and BertScore (BS-rec and BS-F1) (Zhang et al., 2020) using multiple references from the dataset. However, we observe that even a poor quality response can get a high score as per various automated metrics such as BLEU if it matches the tokens in the context or the target. To investigate further, we carry out an experiment where we use the target, context and one of the references as the transition response. An ideal metric would score the reference response high, and give low scores to target and context used as a response.

From Table 2, we observe that most of the standard automated metrics fail to give high scores to the reference response. For example, BLEU assigns higher score to context compared to a human-written reference response. In contrast, the proposed metric TARGET-COHENENCE performs very well in distinguishing between reference response and the distractors.

**Correlation of metrics with human judgements:** Additionally, we investigate how well do various metrics correlate with human ratings of system outputs. To perform this analysis, responses from various methods are judged by crowd-source annotators who rate the smoothness of a response given the dialogue context and the target on a scale of 0 to 1. We use responses sampled from CODA and various baselines, as well as human-written ground truth responses. We collect a total of 440 ratings (ratings and systems outputs will be released) across Otters ID and OOD splits, and report Spearman rank correlation (Spearman, 1961) of the metrics and the ratings. Krippendorff’s alpha for annotation is 0.42. Results, shown in last column of Table 2, depict that most of the standard automated metrics correlate very poorly with human ratings. In contrast, proposed TARGET-COHENENCE achieves a very high correlation score of 0.47.

### 7.4 Results

Next, we discuss evaluation of various system outputs. We report automated metrics as well as human evaluations. Automated metrics measure overlap between model generated outputs and human-written references. Results are summarized in Table 3. We observe that CODA outperforms all the baselines under in-domain as well as out-domain setups of Otters data as per TARGET-COHENENCE. For example, CODA gets a high score of 36.7 as per TARGET-COHENENCE (TC) while the best performing baseline gets only 28.3, demonstrating that the proposed method leads to significant improvements in output quality.

**CODA Ablations:** We observe that: (1) Not using commonsense knowledge (CODA-NOCSKB) leads to large performance drops, highlighting that CODA effectively utilizes commonsense knowledge. (2) Dropping data augmentation leads to a small drop in performance (CODA-NoDA), hinting at relatively small (but still significant) benefit from pretraining the model on re-purposed DailyDialog. (3) CODA-UPPER achieves high scores, highlighting that further improvement in commonsense path generation component can significantly boost the output quality of CODA. (4) Low performance of CODA-KEYWORDS shows the importance of using edges in commonsense paths.

**Human Evaluation:** We conduct human eval-
We present representative outputs from the models we request human annotators to provide their preferences whether the response serves as a smooth transition link between the dialogue context and target, and sensibleness: whether the response makes sense in itself. We present the results of automatic evaluation based on word-overlap and proposed smoothness criteria. CODA outputs are preferred over those of GPT-2 and Multigen on ‘smoothness’ criteria.

### 7.5 Qualitative Analysis

We present representative outputs from the models in Table 5. For CODA, we show the path used in response generation. We notice that GPT-2 and Multigen often tend to either generate simple outputs (e.g. ‘I hate my food’ in the last example) or simply repeat or address either the target or the context (e.g. ‘My pet is the gecko’, ‘Seattle is my favorite city to go to.’). CODA avoids these pitfalls as it is conditioned on generated commonsense paths based on both the context and target entities. However, CODA is susceptible to two issues: 1) Using poor keywords for path generation, and 2) Generation of incorrect paths (e.g. ‘server is a person not desires eat greasy food’ in the last example).

We conduct a human evaluation study to measure the quality of the generated paths. For randomly selected 100 generated responses, we ask annotators to judge 1) Relevance: Is the path relevant and used in the response? and 2) Makes sense: Does the path makes sense? Results reveal that 79% of the paths were judged to be relevant and 76% of the paths were judged to make sense. This indicates that the generated knowledge is good in quality and is used in the response generation. In Appendix B we discuss a human-in-the-loop study for controllability.

### 8 Conclusion

In this work, we propose and evaluate models for target-guided dialogue response generation using explicit commonsense-based bridging paths. We also introduce a reference-less automated metric to evaluate smoothness of a transition response.
References


Ethics Statement

We work on the task of target-guided dialogue response generation. Our proposed models can be used for several useful applications such as providing counselling and creating non-obtrusive recommendations. However, we recognize potential misuse of such models for manipulating users. Our models train on existing datasets such as Otters and DailyDialog, and also leverages external commonsense knowledge resources. As such, our models could potentially inherit biases present in these data sources.

A Additional Method Details

A.1 Clause Identification for Data Augmentation

For target creation, given a dialogue context \( c \) and its response \( r \), we first break the response \( r \) into sentence clauses. For example, given a context “Is my booking complete?” and the response “your reservation is confirmed. now i need your phone number,” we extract a clause \( t \) “i need your phone number” as the target candidate \( t \). For clause extraction we use AllenNlp’s SRL parser \(^1\) which is trained using a BERT-based model (Shi and Lin, 2019) and is based on PropBank (Palmer et al., 2005). It identifies the arguments associated with the predicates or verbs of a sentence predicates (verbs or events) in a sentence and classifies them into roles such as agent, patient and instrument. For the example above, it identifies “need” as a predicate with agent “i” and instrument “your number”.

A.2 Data Augmentation for CODA

We filter data from the dailydialog dataset based on a threshold set to 0.7 for data augmentation. For CODA-NoCSKB model which does not use knowledge paths, the context, target and transition response is used directly in training the CODA-NoCSKB model. But for CODA model which uses the knowledge paths, the dailydialog data is converted to the same format as Otters data, that is, we first do entity detection on the target component of the responses as well as the dialogue context. Then we generate a set of paths for each pair of entities. The CODA model is first trained on dailydialog data with paths and then fine-tuned on the Otters dataset which follows the same knowledge.

\(^1\)github.com/allenai/allennlp

<table>
<thead>
<tr>
<th>Context</th>
<th>Target: i enjoy staring up at the sky.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response 1: i like to spend a lot of my free time with my pet.</td>
<td>(0.99)</td>
</tr>
<tr>
<td>Response 2: I like stargazing outside with my pet.</td>
<td>(0.05)</td>
</tr>
<tr>
<td>Response 3: I like walking with my pet.</td>
<td>(0.01)</td>
</tr>
<tr>
<td>Response 4: My pet is a big star.</td>
<td>(0.02)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Context</th>
<th>Target: i make blogs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response 1: I want to blog about my children.</td>
<td>(0.99)</td>
</tr>
<tr>
<td>Response 2: My family has a lot of babies.</td>
<td>(0.05)</td>
</tr>
<tr>
<td>Response 3: My blogs are very famous.</td>
<td>(0.01)</td>
</tr>
</tbody>
</table>

Table 6: Stress testing the Target-Coherence metric. We show sample responses and TC score for the responses in brackets.

Figure 4: We train a reference-less model-based metric \( \text{TARGET-COHERENCE} \) to score the smoothness of a generate response wrt to dialogue context and target sentence. To train the metric, we synthesize hard negative examples using an ensemble of techniques.

A.3 Target Coherence Metric

In Table 6, we provide examples for stress testing the Target-Coherence metric. TC scores for the responses are shown in brackets. Simply repeating or addressing either the target or context gets a low TC score. In Figure 4 we present an overview of the mechanisms used for generating negative samples for training the Target-Coherence metric.

<table>
<thead>
<tr>
<th>POSITIVE</th>
<th>CONTEXT c</th>
<th>Is my booking complete?</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESPONSE r</td>
<td>Your reservation is confirmed.</td>
<td></td>
</tr>
<tr>
<td>TARGET t</td>
<td>Now i need your phone number</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NEGATIVE</th>
<th>TARGET t'</th>
<th>I am having a problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEGATIVE</td>
<td>CONTEXT c'</td>
<td>What about a draft at 120 days sight ?</td>
</tr>
</tbody>
</table>

A.4 Path Sampling for Response Generator

Since the nodes in ConceptNet are directional, we also add inverse edges during path sampling.

A.5 Training GPT-2 Fudge model

Yang and Klein (2021) proposed a future discriminators based decoding technique. The Fudge discriminator uses a discriminator trained to distinguish good response continuations from the poor ones and guides the GPT2 based decoder towards responses that are coherent to both the source and target sentences. The Fudge discriminator needs
We discard some edge types which are regarded as uninformative and offer little help for our task following Wang et al. (2020). They include RelatedTo, Synonym, Antonym, DerivedFrom, FormOf, EtymologicallyDerivedFrom and EtymologicallyRelatedTo.

### B Human in the loop experiment

#### Can human involvement improve generation?

Our CRG model uses explicit paths generated from the KPG models, which allows human-in-the-loop intervention for finer controllability. To test this hypothesis, we create a model KPG-oneent which is a hybrid version of KPG-wc and KPG-ht model. The model takes an entity $n_k$ given by a user as an input and is trained to generate a path containing that entity. We test this model on a manually created set of target sentences $S$ of size 10 belonging to domains such as healthcare and charity. An example sentence in set $S$ is ‘we should donate to charity’ and we manually curate a set of keywords such as ‘help poor’, ‘make a difference’ and ‘tax deductions’ that are relevant to the target sentence of interest and can guide the knowledge path sampling towards meaningful paths. This data creation took the authors 30 minutes of effort. The data created is shown in Table 7. For 100 random sampled contexts from the Otters dataset, we select a random target sentence from the set $S$ and sample a keyword $k$ from the curated set of keywords of that target. We compare this controllable model with the KPG-ht model that was used for path generation in all our experiments. We find that the TC model favors the KPG-oneent model in 59 percent of cases, confirming that minimal human intervention can improve the quality of generation.

We present sample outputs of the model in Table 8. The input keywords used as intervention are underlined. The paths which use the intervention generate smoother transitions compared to the paths which do not use the intervention.

### C Additional Training Details

We code our models in Pytorch library. We use validation loss to do model selection and use batch size of 10 for GPT-2 models. All GPT-2 models are GPT-small.

#### C.1 Optimizer

We use Adam optimizer with initial learning rate of $1e−4$.

<table>
<thead>
<tr>
<th>Target</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>I need your address</td>
<td>send money; visit; mail; send gift; send coupon</td>
</tr>
<tr>
<td>you should spend time with your friends</td>
<td>don’t be alone; mental health; be happy</td>
</tr>
<tr>
<td>you can try our restaurant</td>
<td>best food; cheapest food; free delivery</td>
</tr>
<tr>
<td>our new recipe is best selling</td>
<td>fat free; healthy; protein; tasty</td>
</tr>
<tr>
<td>I am the best financial advisor</td>
<td>get rich quickly; sound advice; money management</td>
</tr>
<tr>
<td>you should have a positive attitude</td>
<td>mental health; others will help; peace</td>
</tr>
<tr>
<td>we should always avoid fighting</td>
<td>peace; happiness; injury; understand other people</td>
</tr>
<tr>
<td>I want to come to United States</td>
<td>freedom; democracy; money; job; American dream; education</td>
</tr>
<tr>
<td>everyone should get vaccinated</td>
<td>public health; reduce hospital burden; live longer; covid; be safe</td>
</tr>
<tr>
<td>we should donate to charity</td>
<td>help poor; make a difference; tax deductions; feel good; social benefits</td>
</tr>
</tbody>
</table>

Table 7: The set of manually created targets and keyword set used for each target.
Context: i dye my hair.
Target: we should donate to charity.
CODA: I donate to a non-profit that helps people in need.
Path (KPG-oneent): dye hair can be typically done by people desires make a difference is the goal which motivates give assistance has prequisite donate to charity.
CODA: If people who donate are good, they are very good people.
Path (KPG-h): dye hair can be typically done by people desires donate to charity desired by puppy

Context: i have an amazing garden.
Target: you can try our restaurant.
CODA: I made my best food in the garden with tomatoes.
Path (KPG-oneent): garden is a location of grow food motivated by goal best food is desired by person capable of try restaurant
CODA: you can have friends over.
Path (KPG-h): garden is a location of have friends over has prequisite try restaurant desired by puppy

Table 8: Sample data and model outputs for the human intervention experiment. The underlined words are keyword input provided to the model KPG-oneent

C.2 Infrastructure

We use GeForce RTX 2080 GPUs for training models.

D Sample Outputs