Target-Guided Dialogue Response Generation Using Commonsense and Data Augmentation

Anonymous ACL submission

Abstract

Targeted-guided response generation enables dialogue systems to smoothly transition 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, 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 the target, and then uses the identified bridging path to generate transition responses. Additionally, we propose techniques to re-purpose existing dialogue datasets for target-guided generation. Experiments reveal that the proposed techniques outperform various baselines on this task. Finally, we observe that the existing automated metrics for this task correlate poorly with human judgement ratings. We propose a novel evaluation metric that we demonstrate is more reliable for target-guided response evaluation. Our work generally enables dialogue system designers to exercise more control over the conversations that their systems produce.[1]

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 hospitality and education have specific goals to achieve. Prior work has used mechanisms such as emotion labels (Zhong et al., 2019), persona (Song et al., 2019), and politeness (Niu and Bansal, 2018) to control conversations according to the system's agenda. However, such approaches require labelled training data for a set of pre-determined 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. 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.

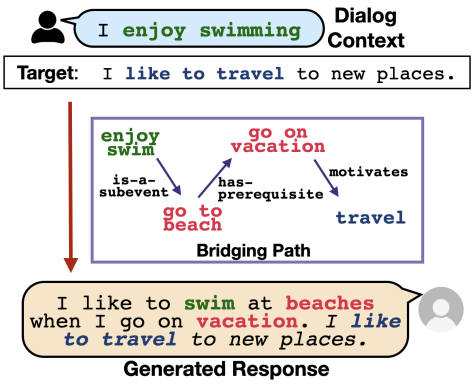

Figure 1: Given a dialogue context and a target sentence, our goal is to generate a dialogue response that smoothly transitions the conversation from context towards the target. Our proposed approach involves identifying a bridging path of entities to link the context and the target.

Existing publicly available dialogue corpora generally consists of free-flow conversations where the speakers move the conversation forward based on the dialogue history alone, absent 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. First, the system needs to balance the

[1] We will release the code publicly

trade-off between coherence with the context while smoothly transitioning towards to the target. Second, the Otters training dataset is relatively small (less than 2000 training instances), making it a low-resource setting. Finally, we show that standard word-overlap metrics are insufficient for this task.

In this work, we propose methods to leverage commonsense knowledge from ConceptNet (Speer et al., 2017a) to improve the quality of transition responses. 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 model's capability to generate logical 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 assign high scores to poor responses just based on high n-gram overlap with reference responses. 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 knowledge 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. In contrast, 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 a response for a given keyword or 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 in dialogue response generation for tasks such as persona-grounded dialogue (Majumder et al., 2020) and open-domain dialogue generation (Ghazvininejad et al., 2018; Hedayatnia et al., 2020; Zhou et al., 2021b). Zhou et al. (2021a) created a dataset focusing on social commonsense inferences in dialogue and Arabshahi et al. (2020) designed a theorem prover for if-then-because reasoning. More broadly, commonsense knowledge has been used in text generation tasks such as story 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.

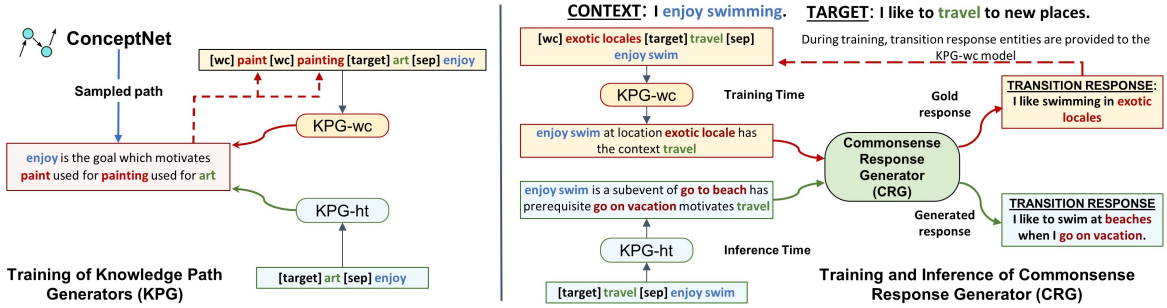

Figure 2: Model illustrations for KPGs - Knowledge Path Generators (left) and CRG - Commonsense Response Generator (Right). Base architecture for all models is GPT-2. Given a path sampled from ConceptNet, KPG-wc learns to predict the path given the head, tail and intermediate entities of the path while KPG-ht learns to predict the path given only the head and tail entities. For the CRG model, during training, a head entity from the context, a tail entity from the target and intermediate entities from the gold transition response are fed into KPG-wc and its output path is used as input to the CRG model. During inference, a head entity from the context and a tail entity from the target are fed into the KPG-ht model. KPG-ht then generates a path with new concepts such as "go on vacation". CRG model conditions on this path for transition response generation.

3 Task Overview

We first formalize the task of target-guided response generation. Given a conversation context c between two speakers A and B, and a target utterance t for speaker B, the task is to generate a transition sentence s which serves as a smooth link between the context and the target. The target is a phrase or a sentence. *Otters* 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 from speaker A, and a target utterance u_b for speaker B, and the task is to generate a transition utterance s for speaker B 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 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)**. We also 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 context c and a target t, a conditional language model learns to predict the transition response s. Target-guided generation can potentially benefit by incorporating commonsense reasoning by identifying rich connections between a pair of entities which enable us to generate logical transition responses connecting the two. Pre-trained language 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 concepts as nodes and has commonsense relationships between them, such as 'IsA' and 'AtLocation'. However, ConceptNet suffers from severe sparsity issues (Malaviya et al., 2020; Bosselut et al., 2019). Therefore, it is not always possible to find the concepts and relationships between context and target concepts. To address the sparsity issue, we develop Knowledge Path Generator (**KPG**), a language model trained on paths sampled from ConceptNet. The model takes a pair of entities or concepts as input and generates a multi-hop path connecting the two. Since the knowledge paths are sampled from a generative model rather than retrieved from a fixed knowledge base, we are no longer limited by the entities and paths present in the ConceptNet knowledge base.

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 (**Figure** 2). 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.

4.1 Commonsense path generator

The KPG models attempts to connect a concept or entity phrase from the context to a concept from the target by creating knowledge paths between them.

Path Sampling: To create training data for the KPG models, we sample paths between entity phrases from ConceptNet using random walks. This step builds upon past work of Wang et al. (2020). Given nodes N and edges E from ConceptNet, 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) n_i, e_i. To sample paths, the random walk begins with a random entity node n_0 and samples a path of random length $k \in \{1, 2, ..., K\}$, where we have set $K = 6$ in this work. To sample paths that are useful for our task, we prevent sampling certain edges types such as *Synonym* (Appendix A.1).

KPG-head-tails (KPG-ht): KPG-ht is a GPT-2 (Radford et al., 2019) based model which is trained to predict a knowledge path p which links a head entity n_h to a tail entity n_t. For a sample path $p = \{n_h, e_0, n_1, e_1, ..., e_{k-1}, n_t\}$ from ConceptNet, the path is formatted into the following sequence "[target] n_t [sep] n_h e_0 n_1 $e_1, \ldots, e_{k-1}$ n_t". KPG-ht is only used during CRG inference where the head entity is extracted from the context and tail entity from the target (Figure 2).

KPG-will-contain (KPG-wc): A large number of possible paths can exist for a given head-tail entity pair. Training the CRG model by conditioning on paths which are irrelevant to the gold transition response might discourage the CRG model from conditioning on the provided commonsense path. Since we do not have gold paths for a response, we instead train a model KPG-wc to generate paths which are more aligned to the gold response by enforcing the generated path to contain entities from the gold response. KPG-wc is trained to predict a path which contains a pre-specified entity set $E_p = \{k_1, ..., k_n\}$ in the generated path by formatting paths sampled from ConceptNet as the following sequence: "[wc] $k1$ [wc] $k2\ldots$ [target] n_t [sep] n_h e_0 n_1 $e_1, \ldots, e_{k-1}$ n_t" (Figure 2). The entity set E_p is a randomly permuted sequence of entities $n_1, n_2, \ldots, n_{k-1}$ from the sampled path. Here "wc" symbolizes "will contain". 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 CRG model training in the next section.

4.2 Response generator

The Commonsense response generator conditions on the commonsense paths generated from the KPG models to generate the 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 using NLTK. We designed simple grammar rules (details in Appendix A.1) to convert phrases to concise forms that match the nodes present in ConceptNet, *e.g.*, "watching the star" is converted to "watch stars".

Sampling and filtering paths: In this step, for 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 to curate 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 gold response, and then apply filtering mechanisms to curate the final set of paths. For training data path sampling, we use the *KPG-wc* model (Figure 2). 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 *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 generate 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) *Perplexity* - We filter out all the paths whose perplexity values (from 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 since repetition often leads to degeneration during decoding. 3) For paths in training data, we filter out paths which contain entities not present in the gold response. The final set of paths 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 (GPT-2 based) 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. For inference, we 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 generates the transition response conditioned on the sequence of c, t, and p.

5 Data Augmentation

The task of target-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. 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, 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 response 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	the restaurant looks authentic european.
Response	the chef trained in florence. the pasta tastes nice here.
SRL Output	predicate = tastes, arguments= the pasta; nice here
Target clause	the pasta tastes nice here.

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*, we run Semantic Role Labelling (SRL) to predict predicate and arguments in a response. For each predicate identified, 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. An example for target creation is shown in Figure 3 (More details about clause identification are in Appendix A.2).

The target creation step does not guarantee that a candidate response transitions smoothly towards the target clause. In the *data filtering* step, we introduce a TARGET-COHERENCE metric to score a transition response in terms of its coherence to the context and smoothness towards the target. 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 transition 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 (Çelikyilmaz et al., 2020). To tackle these challenges, we propose an automatic metric for this task that does not use human references. The proposed metric **TARGET-COHERENCE** is based on a classification model trained to classify a transition response as either

Dataset	Train	Dev	Test
Otters-id	1,929 (693)	1,160 (404)	1,158 (303)
Otters-ood	2,034 (677)	1,152 (372)	1,130 (372)
DailyDialog	11,118	1,000	1,000

Table 1: Overview of the datasets.

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 does not 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 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. An example is shown in Figure 4 of Appendix A.4. We fine-tune a pretrained BERT-base (Devlin et al., 2019) model on these instances with binary cross entropy loss.

7 Experiments

7.1 Datasets

We use two datasets in our experiments. 1) Otters (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. DailyDialog dataset consists of casual conversations between two speakers. In Table 1 we present the number of dialogues in DailyDialog dataset and number of responses in otters, along with number of unique context-target pairs in brackets. Otters dataset consists of multiple responses per context-target pair.

7.2 Baselines for generation

We report results for a number of baselines. We provide complete implementation details of CODA and all baselines in Appendix A and B.

Metric	**Target as response**	**Context as response**	**Reference response**	**Correlation w ratings**
BLEU	15.0	9.9	6.5	-0.11
METEOR	14.0	12.6	13.2	0.01
ROUGE-L	32.3	29.8	26.5	-0.04
BS-rec	38.1	38.9	41.3	0.05
BS-F1	42.8	42.6	38.9	-0.06
TARGET-COHERENCE	10.7	4.0	77.4	0.47

Table 2: We present the metric scores when using the target, context and one of the references as the response. All metrics except for TARGET-COHERENCE score the target and context higher than the reference. TARGET-COHERENCE achieves high correlation with human ratings. Underlined values represent statistically significant result with p-value<0.05.

- **GPT-2:** (Radford et al., 2019) A pretrained GPT–small language model fine-tuned on Otters data. Conditions on the context and target sentences to generate the transition response.
- **GPT2-Fudge** Yang and Klein (2021) uses a discriminator trained to distinguish good response continuations from the poor ones and guides the GPT-2 based decoder towards responses that are coherent to both the source and target sentences.
- **Multigen** (Ji et al., 2020) combines the vocabulary distribution generated by underlying GPT-2 model with a concept distribution from a commonsense knowledge base (ConceptNet).
- **Concept-Predict** leverages a concept prediction strategy from Qin et al. (2020a). The concept is predicted based on closeness to the target.
- **CS-Pretrain** model is pretrained with commonsense paths used for training the KPG models and is based on the commonsense story generation model from Guan et al. (2020).

We report results for following CODA *variants*:

- **CODA-ONLYDA**: CODA variant that uses DailyDialog augmentation and does not use commonsense paths from KPG models in the CRG model.
- **CODA-NODA**: CODA trained without additional data from DailyDialog.
- **CODA-NOEDGE** CODA variant that uses only entities and no edges from the path.
- **CODA-NOALIGN**: variant that relies on only KPG-ht for training and inference. Does not select paths based on alignment with responses.
- **CODA-KBPATH**: variant that retrieves paths directly from ConceptNet using the algorithm proposed in Lin et al. (2019).
- **CODA-Upper** Upper bound for CODA which uses paths inferred from the gold responses using the KPG-wc keywords model during inference.

	In-Domain					Out-Of-Domain				
	BLEU	METEOR	ROUGE-L	BS-rec	TC	BLEU	METEOR	ROUGE-L	BS-rec	TC
GPT-2	3.4	11.9	23.9	35.4	26.7	3.0	10.8	22.2	35.0	29.7
GPT2-Fudge	3.4	12.4	24.4	36.1	28.3	3.4	11.1	23.0	35.1	29.6
Multigen	6.2	12.5	28.1	40.0	27.8	4.9	11.6	26.0	36.7	30.8
Concept-predict	3.3	12.3	28.5	38.1	28.3	3.7	11.6	23.1	35.9	26.3
CS-Pretrain	2.8	11.1	23.2	35.2	21.5	2.8	10.2	21.2	33.0	22.0
CODA	5.0	12.6	25.9	38.0	**36.7**	4.6	11.5	24.3	35.5	**37.9**
CODA-ONLYDA	4.0	12.4	24.4	37.5	32.7	3.1	11.1	22.7	35.3	33.2
CODA-NODA	4.4	12.3	25.1	37.8	35.7	4.5	11.6	24.4	35.4	36.0
CODA-NOEDGE	4.2	12.0	25.0	37.4	33.7	4.0	11.8	24.2	35.4	35.9
CODA-NOALIGN	3.7	12.4	25.5	38.5	32.1	3.2	11.2	22.8	35.6	31.2
CODA-KBPATH	3.6	12.5	24.9	38.6	33.9	3.6	11.4	24.1	35.9	33.0
CODA-UPPER	8.3	18.1	32.6	44.4	47.9	7.5	17.9	30.7	42.7	45.4
Human	6.5	13.1	26.5	41.3	77.4	4.9	12.3	24.0	37.6	77.3

Table 3: We present the results of automatic evaluation based on word-overlap and proposed TARGET-COHERENCE. CODA outperforms all the baselines for most of the metrics. We also present results for CODA's model ablations.

7.3 Evaluation Metrics

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). Evaluation is carried out using multiple references from the test set. Word-overlap metrics do not correlate well with human judgements (Liu et al., 2016). Additionally, we observe that on this task, even a poor transition response can get a high score on reference-based metrics if it has high overlap with the context or the target. 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. In Table 2, reference-based metrics assign higher scores to target and context sentences used as responses compared to human-written responses. In contrast, TARGET-COHERENCE assigns high scores to reference responses and low scores to target and context sentences.

Correlation of metrics with human judgements: We investigate how well do the metrics correlate with human ratings of system outputs. To perform this analysis, responses from CODA, baselines, as well as reference responses are judged by crowdsource annotators who rate the smoothness of a response given the dialogue context and the target on a scale of 0 to 1 (Appendix C). We collect a total of 440 ratings 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. Ratings and systems outputs will be released. Results, shown in last column of Table 2, depict that most standard automated metrics correlate poorly with human ratings, while the, proposed TARGET-COHERENCE achieves a very high correlation score of 0.47.

7.4 Results

In this section we present the automatic and human evaluation results. Automated metric results are summarized in Table 3. Although reference-based metrics are lexically biased (subsection 7.3), we still report their scores. We observe that CODA outperforms all the baselines under in-domain (ID) as well as out-of-domain (OOD) setups of Otters data as per TARGET-COHERENCE (TC) score. For example, CODA gets a high TC score of 36.7 (ID) and 37.9 (OOD) while the TC scores of the closest baselines GPT2-Fudge, Multigen and Concept-predict are in the range of 28-31, demonstrating that the proposed method leads to significant improvements in response quality. However, CODA is far from reaching human performance (TC 77.4).

CODA Ablations: We observe that: (1) Not using commonsense knowledge (CODA-ONLYDA) 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 using data augmentation. (3) Low performance of CODA-NOEDGE shows the importance of using edges in commonsense paths. (4) Not aligning and selecting paths based on their relevance to responses during CRG training (CODA-NOALIGN) leads to a high drop in performance. (5) CODA outperforms CODA-KBPATH by 8% (ID) and 14.5% (OOD). This improved performance can be attributed to the generalizability of entities and paths generated from the KPG models. (6) CODA-UPPER achieves high scores, highlighting that further improvement in commonsense path generation component can sig-

Criteria	Models	Win	Lose	Tie
Smooth	CODA vs GPT-2	37.5	31.6	31.0
	CODA vs Multigen	32.3	22.8	44.8
Sensible	CODA vs GPT-2	22.0	21.3	56.7
	CODA vs Multigen	25.8	25.6	48.6
Informative	CODA vs GPT-2	32.3	27.3	40.4
	CODA vs Multigen	35.5	27.8	36.7

Table 4: Human evaluation through pairwise comparison between CODA and baselines. CODA is preferred in smoothness and informativeness criteria while being comparably sensible.

nificantly boost the output quality of CODA.

Human Evaluation: We conduct human evaluations on Amazon Mechanical Turk to evaluate the quality of generated transition responses. Annotators are requested to evaluate the transition response on following criteria: (1) *Smooth*: rate whether the response serves as a smooth transition between the dialogue context and target. (2) *Sensible*: whether the response makes sense in itself i.e. it is grammatical and logically coherent. (3) *Informative*: how much informative content a response carries. Human annotators compare (or mark as a tie) responses from two models. We collect two annotations for 100 randomly selected data points from the test outputs. Results in Table 4 demonstrate that CODA outputs are preferred over the baselines on 'Smooth' and 'Informative' 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.') which leads to high BLUE and METEOR scores, but low TC scores. 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 irrelevant paths (*e.g.* 'server is a person not desires greasy food' in the last example).

Path quality: 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. Thus in aggregate, the generated knowledge is good in quality, and is used in the generated response.

Context: i like the sand on my feet *Target*: my puppy is called georgie.
GPT-2: My mom likes the water. *Multigen*: My pet is the gecko. *CODA*: My dog walks along the beach with sand. *CODA-Path*: sand is at location beach belongs to walk is desired by puppy
Context: my favorite city is seattle. *Target*: i ride my bicycle everywhere.
GPT2: Seattle is my favorite city to go to *Multigen*: So what do you do when you go to the seattle *CODA*: I bought my bicycle from a bike shop in seattle. *CODA-Path*: favorite city is the location which has bicycle shop is a dependency of ride bicycle
Context: i am a server at a food place. *Target*: i eat greasy foods.
GPT-2: I eat healthy foods at restaurants. *Multigen*: I hate my food. *CODA*: I am a server, but I don't want to eat too much. *CODA-Path*: server is a person not desires eat greasy food

Table 5: Sample representative model outputs.

Path novelty: We analyzed the paths generated by CODA which were judged as sensible by human annotators and found that 26.8% of entities in the paths were not found in ConceptNet. This include entities such as 'favorite food', 'pet kitten', 'single kid' and 'online class'. Thus, the actual paths from the ConceptNet might not be able to cover a large fraction of head/tail entities. Furthermore, 81% of sensible paths are novel and do not exist in ConceptNet. For example, even though the path 'eat motivates go to restaurant has subevent dinner is the location for bread' exist in ConceptNet, the path 'eat motivates go to restaurant has subevent dinner is the location for pizza' does not exist in ConceptNet. Thus we show that CODA can generalize to new entities and paths.

In Appendix D we discuss a human-in-the-loop study for controllability.

8 Conclusion

In this work, we propose and evaluate models for target-guided response generation using explicit commonsense bridging paths. We also introduce an automated metric to evaluate smoothness of a transition response. We showed that our model generates more smooth and informative outputs through automatic and human evaluation. Furthermore, it allows for more interpretable results. Going forward, we envision a model which could combine target and non-target guided dialogue planning.

References

Forough Arabshahi, Jennifer Lee, Mikayla Gawarecki, Kathryn Mazaitis, Amos Azaria, and Tom Mitchell. 2020. Conversational neuro-symbolic commonsense reasoning. *arXiv preprint arXiv:2006.10022*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2492–2501. Association for Computational Linguistics.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *AAAI*.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108.

Jian Guan, Yansen Wang, and Minlie Huang. 2019a. Story ending generation with incremental encoding and commonsense knowledge. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6473–6480. AAAI Press.

Jian Guan, Yansen Wang, and Minlie Huang. 2019b. Story ending generation with incremental encoding and commonsense knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6473–6480.

Prakhar Gupta, Yulia Tsvetkov, and Jeffrey P. Bigham. 2021. Synthesizing adversarial negative responses for robust response ranking and evaluation. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3867–3883. Association for Computational Linguistics.

Behnam Hedayatnia, Karthik Gopalakrishnan, Seokhwan Kim, Yang Liu, Mihail Eric, and Dilek Hakkani-Tur. 2020. Policy-driven neural response generation for knowledge-grounded dialog systems. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 412–421, Dublin, Ireland. Association for Computational Linguistics.

Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 725–736. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. 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 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yanxiang Ling, Fei Cai, Xuejun Hu, Jun Liu, Wanyu Chen, and Honghui Chen. 2021. Context-controlled topic-aware neural response generation for open-domain dialog systems. *Inf. Process. Manag.*, 58(1):102392.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An

empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126.

Bodhisattwa Prasad Majumder, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Julian J. McAuley. 2020. Like hiking? you probably enjoy nature: Persona-grounded dialog with commonsense expansions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9194–9206. Association for Computational Linguistics.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. Commonsense knowledge base completion with structural and semantic context. *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics*, 6:373–389.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

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.

Jinghui Qin, Zheng Ye, Jianheng Tang, and Xiaodan Liang. 2020a. Dynamic knowledge routing network for target-guided open-domain conversation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8657–8664. AAAI Press.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D. Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020b. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 794–805. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ananya B Sai, Akash Kumar Mohankumar, and Mitesh M Khapra. 2020. A survey of evaluation metrics used for nlg systems. *arXiv preprint arXiv:2008.12009*.

Karin Sevegnani, David M. Howcroft, Ioannis Konstas, and Verena Rieser. 2021. Otters: One-turn topic transitions for open-domain dialogue. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2492–2504. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Haoyu Song, W. Zhang, Yiming Cui, Dong Wang, and T. Liu. 2019. Exploiting persona information for diverse generation of conversational responses. In *IJCAI*.

Charles Spearman. 1961. The proof and measurement of association between two things. *Appleton-Century-Crofts*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017a. Conceptnet 5.5: An open multilingual graph of general knowledge.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017b. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric P. Xing, and Zhiting Hu. 2019. Target-guided open-domain conversation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5624–5634. Association for Computational Linguistics.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2017. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. *arXiv preprint arXiv:1701.03079*.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020. Connecting the dots: A knowledgeable path generator for commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4129–4140, Online. Association for Computational Linguistics.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804, Florence, Italy. Association for Computational Linguistics.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots. *arXiv preprint arXiv:2010.07079*.

Kevin Yang and Dan Klein. 2021. FUDGE: controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3511–3535. Association for Computational Linguistics.

Pengcheng Yang, Lei Li, Fuli Luo, Tianyu Liu, and Xu Sun. 2019. Enhancing topic-to-essay generation with external commonsense knowledge. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2002–2012. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 654–664. Association for Computational Linguistics.

Peixiang Zhong, Yong Liu, Hao Wang, and Chunyan Miao. 2021. Keyword-guided neural conversational model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14568–14576.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. An affect-rich neural conversational model with biased attention and weighted cross-entropy loss. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7492–7500.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4623–4629. International Joint Conferences on Artificial Intelligence Organization.

Pei Zhou, Karthik Gopalakrishnan, Behnam Hedayatnia, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. 2021a. Commonsense-focused dialogues for response generation: An empirical study. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 121–132, Singapore and Online. Association for Computational Linguistics.

Pei Zhou, Behnam Hedayatnia, Karthik Gopalakrishnan, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. 2021b. Think before you speak: Learning to generate implicit knowledge for response generation by self-talk. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 251–253, Online. Association for Computational Linguistics.

Asli Çelikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *ArXiv*, abs/2006.14799.

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. Xu et al. (2020) provides a review of recent methods that try to mitigate safety issues in open-domain dialogue generation which can be utilized for our task.

A Implementation Details for CODA

A.1 Training Details for CODA

Model training: We code our models using Pytorch and Huggingface [2] library. We use validation loss to do model selection. The KPG-wc, KPG-ht and CRG models are all based on GPT-2 small architecture. We use batch size of 10 for GPT-2 models. We use Adam optimizer with initial learning rate of $1e-4$. We use GeForce RTX 2080 GPUs for training models. All existing code used and datasets were CC-BY 4.0 or open sourced by original authors.

Decoding paths and responses: For decoding paths using the KPG models, we use temperature of 0.7 and nucleus sampling with top-p set to 0.9. We use the same decoding strategy and hyperparameters for decoding responses using CRG model.

Concept Extraction: Entities need to be extracted from the context, target and response to generate and align paths using the KPG models. For any given sentence s, we first extract the set of noun and verb phrases from the sentence using NLTK. We design some simple grammar rules to convert some phrases to a more concise forms that are similar to the kinds of nodes present in ConceptNet,e.g., "watching the star" is converted to "watch stars". We use NLTK's POS tagging combined with the following grammar rules: (1) Nouns and Adjectives, terminated with Nouns <NN.*|JJ>*<NN.*> (2) Verb and verb phrases <RB.?>*<VB.?>*<JJ>*<VB.?>+<VB>?. We normalize the verbs using NLTK. The final set of entities consist of the noun and verb phrases. We exclude certain phrases such as "today", "enough" which are sometimes incorrectly detected as entities.

Sub-selecting entity pairs during training of CRG model: For every context-target pair, we have n number of pair of head-tails entities. 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 topD pairs of entities. The value of top D is selected based on validation performance and comes out typically between 1-3.

Knowledge graph details: The number of nodes in the ConceptNet resource we have used[3] is 382226. We perform random walks on the graph with paths of length from 1 to 6 and get a total of 3883671 number of paths.

Edges in the knowledge path: We discard some edge types which are regarded to be uninformative and offer little help for our task folowing Wang et al. (2020). They include RelatedTo, Synonym, Antonym, DerivedFrom, FormOf, EtymologicallyDerivedFrom and EtymologicallyRelatedTo. Since the nodes in ConceptNet are directional, we also add inverse edges during path sampling. For example the path "ecosystem <– PartOf <– organism" can be sampled as "ecosystem _isPartOf organism" where the underscore indicates a reverse edge.

A.2 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 [4] 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".

[2] https://huggingface.co/

[3] https://github.com/wangpf3/Commonsense-Path-Generator

[4] github.com/allenai/allennlp

A.3 Data Augmentation for CODA

We filter data from the dailydialog dataset based on a threshold set to 0.7 for data augmentation. This threshold was selected using emperical performance of thr CODA model. For CODA-ONLYDA model which does not use knowledge paths, the context, target and transition response is used directly in training the CRG decoder of CODA-ONLYDA 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 the dialogue context. Then we generate a set of paths for each pair of entities. The CODA model is first trained on paths from the filtered dailydialog data and then fine-tuned on the Otters dataset which follows the same knowledge path format. The maximum dialogue history length is set to 2 for dailydialog dataset.

A.4 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. For example the response "I like stargazing outside" is not a smooth transition and gets a low TC score, while "I like stargazing outside with my pet" is a smooth transition and gets a high TC score. In Figure 4 we present an overview of the mechanisms used for generating negative samples for training the Target-Coherence metric. For negative examples, 1) Given gold response r, and context c, we sample a random negative target t', which creates a response which does not transition towards the target t, 2) Given gold response r, and target t, we sample a random negative context c', which creates a response which is not coherent to the context c, 3) Given gold context c, and target t, we either sample a random negative response r' or generate a response r' conditioned on random c' or t', which creates a response which does not transition to target t or is coherent to context c.

Context: i enjoy staring up at the sky. Target: i like to spend a lot of my free time with my pet. Response 1: I like stargazing outside with my pet. (0.99) Response 2: I like stargazing outside. (0.05) Response 3: I like walking with my pet. (0.01) Response 4: My pet is a big star. (0.02)
Context: i make blogs. Target: i have a large family with babies. Response 1: I want to blog about my children.(0.99) Response 2: My family has a lot of babies. (0.05) Response 3: My blogs are very famous. (0.01)

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

POSITIVE Gold c,r,t	CONTEXT c	the restaurant looks authentic european.
	RESPONSE r	the chef trained in florence. the pasta tastes nice here.
	TARGET t	the pasta tastes nice here.
NEGATIVE Random t' with gold r,c	TARGET t'	i love to drive my car.
NEGATIVE Random c' with gold r,t	CONTEXT c'	i enjoy computers and phones.
NEGATIVE Random r' with gold c,t	Response r'	there is no parking here.

Figure 4: We train a reference-less model-based metric 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.

B Training Details of Baselines

Training GPT-2 Fudge model Yang and Klein (2021) proposed a future discriminator 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 positive and negative sample data for training. We train the discriminator to distinguish a good response from a bad (not coherent to target or context). The input to train the discriminator (a LSTM model) is the concatenation of the context sentence, followed by the target sentence and finally the tokens of a response r with tokens k. The discriminator then learns to predict 1 if the next token in the response at position k belongs to the gold response or 0 if the token is a random one. We train the Fudge discriminator by preparing negative instances using the same techniques we use to train the Target-Coherence model - sampling random negative responses, responses coherent to the context but not to the target, and responses coherent to the target but not to the context.

Training CS-Pretrain model The model is based on the commonsense story generation model from Guan et al. (2020) We create training data for the CS-Pretrain model by using the same sampled paths we use for training the KPG-wc model. The paths are converted into textual format by convert-

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

Table 7: The set of manually created targets and keyword set used for each target.

ing edges into text sequences. The model is only pretrained with general commonsense paths and then fine-tuned on Otters dataset in a manner similar to the GPT-2 baselines (i.e. without paths). Our experiments show that pretraining with commonsense model does not help with target-guided task, probably since the task needs target conditional commonsense and general commonsense knowledge only confuses the model during decoding.

Training Concept-Predict leverages a concept prediction strategy from Qin et al. (2020a). The input to the model is the context and target and it predicts a single concept based on closeness to the target. The concept is then fed as an input to the CRG model along with the context and target sentences.

Training CODA-OnlyDA: CODA variant that uses Dailydialog augmentation and does not use commonsense paths from KPG models in the CRG model. Therefore the model consists of only a CRG model (no KPG models) which take the context and target sentences as inputs.

Training CODA-NoEdge CODA variant that uses only entities and no edges from the path. For example the path "favorite city is the location which has bicycle shop is a dependency of ride bicycle" is converted to "favorite city bicycle shop ride bicycle", which is fed as input to the CRG model.

Training CODA-NoAlign: variant that relies on only KPG-ht for training and inference. Does not select paths based on alignment with responses. The paths used during training the CRG model come from KPG-ht instead of KPG-wc.

Training CODA-KBPath: variant that samples paths directly from ConceptNet using the algorithm proposed in Lin et al. (2019). Given a pair of context and target concept, we use their algorithm to sample an actual path directly from ConceptNet. The model is pretrained on Dailydialog augmented data and fine-tuned on Otters with the sampled paths from ConceptNet. The model suffers from missing entities and missing links between entities in ConceptNet which is solved by CODA.

Context: i dye my hair. Target: we should donate to charity.
Path (KPG-oneent): hair belongs to people motivated by give assistance has prequisite donate to charity. CODA-controlled: I donate my hair to a non-profit that *helps people in need*. Path (KPG-ht): hair belongs to people desires donate to charity CODA: People who donate are very good people.
Context: i have an amazing garden. Target: you can try our restaurant.
Path (KPG-oneent): garden is a location of grow food motivated by goal best ingredients is desired by person capable of try restaurant CODA-controlled: My restaurant uses the *best ingredients* from the garden. Path (KPG-ht): garden is a location of have friends over has prerequisite try restaurant CODA: you can have friends over.

Table 8: Sample data and model outputs from the human-in-the-loop experiment. The underlined words are keyword inputs provided to the model KPG-oneent. The italicised words in the CODA controlled outputs are phrases are generated based on the input keywords.

C Human Ratings Collection

We present the Amazon mechanical turk interface for human ratings collection in Figure 5. THe workers were first shown instructions about the task with definitions and examples for all rating criteria. We paid the workers an average of 15 per hour. We set the qualification condition as 1000 HITS completed, 95% or more approval rate and location as native english speaking countries.

D Human-in-the-loop Experiment

Can human involvement improve generation? Our CRG model uses explicit paths generated from the KPG models, which not only provides interpretability, it also allows human-in-the-loop intervention for finer controllability. To test this hypothesis, we create a model KPG-oneent which is a

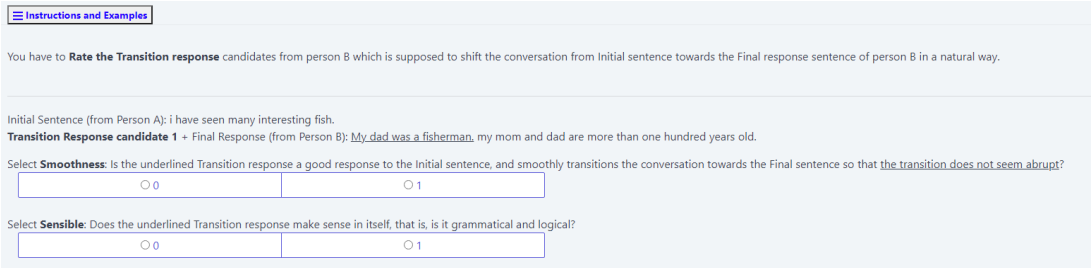

Figure 5: Amazon mechanical turk interface for human ratings collection

hybrid version of KPG-wc and KPG-ht model. The model takes a single 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. The data created is shown in Table 7. An example sentence in set S is 'we should donate to charity' and we manually curate a set of keywords such as 'help poor', 'give assistance' 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. 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 TARGET-COHERENCE metric favors the KPG-oneent model in 59 percent of cases, confirming that even minimal human intervention in the form of domain relevant keywords 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 keyword intervention generate smoother transitions compared to the paths which do not use the keyword intervention.