

Context Matters in Semantically Controlled Language Generation for Task-oriented Dialogue Systems

Ye Liu¹, Wolfgang Maier², Wolfgang Minker³ and Stefan Ultes⁴

^{1,2,4}Mercedes-Benz AG, Sindelfingen, Germany

^{1,3}Ulm University, Ulm, Germany

¹ye.y.liu@daimler.com

^{2,4}{wolfgang.mw.maier, stefan.ultes}@daimler.com

³wolfgang.minker@uni-ulm.de

Abstract

This work combines information about the dialogue history encoded by pre-trained models with a meaning representation of the current system utterance to realize contextual language generation in task-oriented dialogues. We utilize the pre-trained multi-context ConveRT model for context representation in a model trained from scratch; and leverage the immediate preceding user utterance for context generation in a model adapted from the pre-trained GPT-2. Both experiments with the MultiWOZ dataset show that contextual information encoded by pre-trained models improves the performance of response generation both in automatic metrics and human evaluation. Our presented contextual generator enables higher variety of generated responses that fit better to the ongoing dialogue. Analysing the context size shows that longer context does not automatically lead to better performance, but the immediate preceding user utterance plays an essential role for contextual generation. In addition, we also propose a re-ranker for the GPT-based generation model. The experiments show that the response selected by the re-ranker has a significant improvement on automatic metrics.

1 Introduction

In a conversation, speakers are influenced by previous utterances and tend to adapt their way of speaking to each other (Dušek and Jurcicek, 2016; Reitter et al., 2006). Furthermore, generating the responses that fit well to dialogue context facilitates successful conversation and strengthens the user’s impression of Spoken Dialogue Systems (SDSs). Several previous works (Dušek and Jurcicek, 2016; Kale and Rastogi, 2020; Sankar et al., 2019) have explored the impact of previous dialogue information on the generated language in task-oriented dialogue. However, how to efficiently infuse the

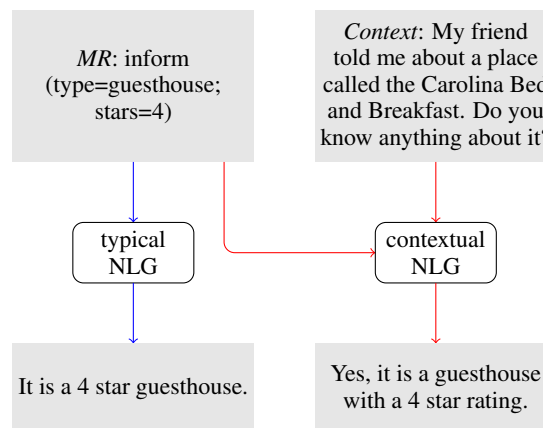


Figure 1: An example of contextual NLG (red part) compared with typical NLG (blue part) in our experiments. (More examples with multi-turn context please find in the Appendix)

dialogue context into a semantically controlled generator for improving the contextual interactive experience is still challenging. Such as, the contextual generator proposed in Dušek and Jurcicek (2016) has no big improvement without the help of an additional re-ranker. The empirical study in Sankar et al. (2019) demonstrated that both recurrent and transformer-based seq2seq model can not effectively consider previous dialogue history for generation. In this work, we propose two contextual generators, which both utilize pre-trained models to encode dialogue context. And the experiment results show that context does matter in semantically controlled task-oriented Natural Language Generation (NLG).

The function of NLG in task-oriented SDS is to generate meaningful output in the form of natural language with the guidance of meaning representation (MR). The MR is a formalism of response semantics and generally represents a dialogue action (DA), such as *inform* or *request*, along with one or more slots and their values (See the MR in

Figure 1). However, this typical NLG only takes the input MR into account and has no clue on how to adapt to the dialogue history. This results in coarse and flat responses (see the blue part in Figure 1). To enable the contextual interaction in SDS, task-oriented NLG should not only contain the desired MR information, but also have access to the dialogue history. The example shown in Figure 1 taken from our experiments: compared with non-contextual generation, the response with context guidance adapts better to the preceding dialogue and is more like a sentence from a real human. In addition, the contextual NLG models are prone to generate more diverse responses.

In summary, the main contributions of this paper are as follows:

- To leverage the contextual nature of the multi-turn dialogue, we utilize the pre-trained multi-context ConveRT (Henderson et al., 2020) to encode dialogue history for the contextual generator. These contextual embeddings are then forwarded to the Semantically Controlled LSTM (SC-LSTM) generator (Wen et al., 2015). The ConveRT initialized SC-LSTM is called *CSC-LSTM* for short in this paper. With the powerful multi-context encoding of ConveRT, we also analyse the impact of variable context size on *CSC-LSTM*. To the best of our knowledge, we are the first to utilize the pre-trained conversational model ConveRT for contextual generation in task-oriented dialogue system.
- We leverage only the immediate preceding user utterance for contextual generation. Adapted from GPT-2 (Radford et al., 2019), the user utterance and DA guide the contextual generation as context and semantic information respectively. We call Contextually and Semantically Conditioned GPT *CSC-GPT* for short in this paper. The experiments of GPT-based contextual model show that generation benefits from dialogue context, even if only immediate preceding user utterance is taken into account.
- We propose a BERT-based (Devlin et al., 2019) Re-ranker (*BERT-R*) for the *CSC-GPT* generator, to select system response that fit better to the user utterance. Given the top 5 generations of *CSC-GPT*, several automatic

scores are regarded as contextual basis between user utterance and system response to train on a multiple regression task adapted from BERT. Experiments show that the re-selected generation has a significant improvement on the performance scores.

The remainder of this paper is structured as follows: Section 2 shows the related works of our research. Section 3 introduces the dataset and the automatic metric scores which are used in this work. Section 4 describes our proposed three models: *CSC-LSTM*, *CSC-GPT* and *BERT-R*, as well as the experiment details. Section 5 shows the experiment results of all models in automatic metrics and human evaluation. The last Section 6 concludes and outlines future research.

2 Related Works

For task-oriented NLG, semantically controlled neural models play a significant role. Wen et al. (2015) introduced a semantically conditioned model by adding an additional semantic cell in LSTM to control the DA, which is defined as the combination of intent and slot-value pairs, for generation. Tseng et al. (2018) improved the RNN-based generator by considering latent information using the semantically conditional variational autoencoder architecture. As the major advantage and superior performance of pre-trained LMs (Devlin et al., 2019; Radford et al., 2019), Peng et al. (2020b) proposed a semantically controlled generation model based on GPT-2; Chen et al. (2020) and Peng et al. (2020a) presented an end-to-end task-oriented SDS based on the pre-trained GPT-2. Even though there has been plenty of works on semantically guided NLG, most approaches fail in utilizing information of the preceding interaction.

Dušek and Jurcicek (2016) stood out as they extended the idea of NLG from MRs by adding *one* preceding user utterance to their recurrent encoder. However, we are more interested in the influence of bigger context sizes in *CSC-LSTM* contextual generation. And the model in Dušek and Jurcicek (2016) was not able to show any improvement for contextual generation without an additional n -gram match re-ranker. However, our proposed contextual generators outperform both the baselines even without re-ranker and the re-ranker in our work can further highly improve the generation performance on automatic metrics. Sankar et al. (2019) made an empirical study to understand how models use the

available dialog history for generation and found that both recurrent and transformer-based seq2seq model can not consider previous dialogue history effectively. However, the dialogue history in our work is encoded by pre-trained models and experiments show that the generation in task-oriented dialogue benefits from dialogue history. [Kale and Rastogi \(2020\)](#) also examined the role of context and demonstrated that the generation benefits from the dialogue history. While their approach highly relies on manually pre-defined templates which are costly to create, in this work, all responses are directly generated without the need of any templates.

3 Dataset and Automatic Metrics

In our work, the automatic metric scores are not only used for performance evaluation, but also used for the *BERT-R* training in Section 4.3. Hence, we introduce the dataset and the automatic metrics beforehand in this section.

3.1 MultiWOZ dataset

The original MultiWOZ ([Budzianowski et al., 2018](#)) dataset is a fully-labeled collection of human-human written conversations spanning over multiple domains and topics. It contains over 10,000 dialogues spanning 8 domains, namely: Restaurant, Hotel, Attraction, Taxi, Train, Hospital, Bus, and Police. The test and validation sets contain 1,000 examples each for performance comparison. The MultiWOZ 2.1 ([Eric et al., 2020](#)) and MultiWOZ 2.2 ([Zang et al., 2020](#)) both fix some dialogue state annotation errors and dialogue utterances, resulting in an improved version of the original MultiWOZ.

The MultiWOZ 2.1 is used for evaluation of *CSC-LSTM* in Section 4.1 in accordance with related work. The MultiWOZ 2.1 and 2.2 are both used additionally for evaluating *CSC-GPT* and *BERT-R* in Section 4.2 and 4.3.

3.2 Automatic metrics

The following metrics are used for performance comparison in Section 5 and several are applied for *BERT-R* training as target score in Section 4.3.

3.2.1 N-gram matching metrics

BLEU-4 is the 4-gram BLEU score ([Papineni et al., 2002](#)), which is the most widely used metric score for evaluating the performance of language generation and machine translation. In this work, BLEU-4 is computed for multiple values of $n = (1, 2, 3, 4)$ with weights $(0.25, 0.25, 0.25, 0.25)$ respectively

and the scores are averaged geometrically. A smoothing function is used to avoid that no n -gram overlaps are found.

The target signal $\tau_{\text{BLEU-4}}$ in Section 4.3 represents the BLEU-4 score between system generated response and gold reference in *BERT-R*.

Meteor ([Banerjee and Lavie, 2005](#)) is a kind of weighted F-score based on mapping unigrams and also computes a penalty function for incorrect word order. [Lavie and Agarwal \(2007\)](#) demonstrated that Meteor score has high correlation with human ratings.

The target signal τ_{Meteor} in Section 4.3 represents the Meteor score between system generation and gold reference in *BERT-R*.

3.2.2 Machine learned metric

BERTScore ([Zhang et al., 2019](#)) is a machine learned automatic evaluation metric for text generation that has shown a high correlation with human judgments. BERTScore leverages the pre-trained contextual embeddings from variants of BERT ([Devlin et al., 2019](#)) and matches words in candidate and reference sentences by cosine similarity. Moreover, BERTScore computes precision, recall, and F1 measure¹. [Zhang et al. \(2019\)](#) showed that the Roberta ([Liu et al., 2019](#)) large model has the best-performing results for English tasks. So the roberta-large model² is used for computing BERTScore in this work.

The target signals $\tau_{\text{BERT}_{\text{pre}}}$, $\tau_{\text{BERT}_{\text{rec}}}$, $\tau_{\text{BERT}_{\text{f1}}}$ in Section 4.3 represent the precision, recall and F1 of BERTScore between system response and gold reference respectively in *BERT-R*.

3.2.3 Other metrics:

ConveRT cosine similarity: ConveRT ([Henderson et al., 2020](#)) is a light-weight conversational model pre-trained on the large Reddit conversational corpus ([Henderson et al., 2019](#)). It provides powerful representations for conversational data and can be used as a response ranker by comparing the cosine similarity between user utterance and multiple responses. In this work, we not only utilize the pre-trained ConveRT for context embedding in *CSC-LSTM*, but also for a target score in *BERT-R*.

The target signal $\tau_{\text{ConveRT-cs}}$ in Section 4.3 means

¹Only F1 score, which represents a reasonable balance between recall and precision, is shown in Table 1, Table 2, Table 4, Table 5 and Figure 2 for performance comparison.

²https://github.com/Tiiiger/bert_score

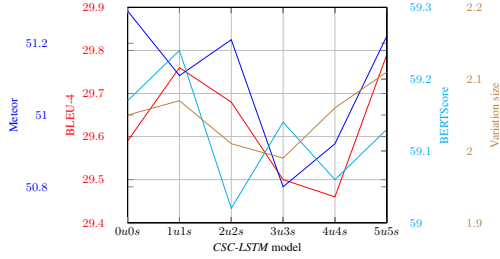


Figure 2: The curves of BLEU-4, Meteor, BERTScore and Variation size over CSC-LSTM model with different context size (the exact value of metrics please find in Table 1) show similar tendency: the both ends of every curve have better performance than the inner part.

the cosine similarity of ConveRT embedding between user utterance and system response in *BERT-R*.

Variation size measures the variation of the generated system responses, i.e., how many different realisations are generated for one DA on average. The variation size results computed on the full MultiWOZ test data are shown in the left part of the results column (/) while the right part shows the variation size computed only over the instances of the test data with DA that appear more than once, i.e., where variation can actually occur.

4 The Proposed Models and Experiment Details

In this section, our proposed models (two contextual generators, one response re-ranker) and corresponding experiment details are introduced.

4.1 ConveRT initialized SC-LSTM: CSC-LSTM

We train *CSC-LSTM*³ on the basis of the SC-LSTM (Wen et al., 2015), where a semantic control cell encodes DA into an one-hot embedding to guide the task-oriented generation that is oblivious about any dialogue history. In our proposed *CSC-LSTM*, we apply the pre-trained multi-context ConveRT⁴ for encoding the dialogue history and the contextual embedding is forwarded to initialize the SC-LSTM generator. Before initialization in *CSC-LSTM*, a non-linear transformation⁵ is applied, which is shown in Eq. 1, to project the ConveRT

³The architecture of *CSC-LSTM* is shown in Appendix.

⁴<https://github.com/davidalami/ConveRT>

⁵Using the same hidden size as the dimension of the ConveRT embedding, i.e., *CSC-LSTM* (hidden size 512) without project function d_0 in Eq. 1 results in worse performance.

embedding into the SC-LSTM decoder space:

$$h_0 = \tanh(\mathbf{W}C_e + \mathbf{b}). \quad (1)$$

$h_0 \in \mathbb{R}^{d_g}$ is the SC-LSTM decoder initial recurrent state, $C_e \in \mathbb{R}^{d_c}$ is the ConveRT context embedding and $\mathbf{W} \in \mathbb{R}^{d_g \times d_c}$ projects the context level embedding into the decoder space. The \mathbf{W} and \mathbf{b} are learnable parameters during the *CSC-LSTM* training.

Experiment details of *CSC-LSTM*: For *CSC-LSTM*, the SC-LSTM will be used as baseline without additional context information. This means, for each utterance generation, the hidden state is initialized with zeros in SC-LSTM. The MultiWOZ 2.1 (Eric et al., 2020) dataset is used for SC-LSTM and *CSC-LSTM* generation models.

The multi-context ConveRT embedding dimensionality is 512, hence, the d_c in Eq. 1 is 512 for *CSC-LSTM* training. In order to ensure a fair comparison, we set the same hyper-parameters for SC-LSTM and *CSC-LSTM*: the hidden size to 300 (the d_g in Eq. 1), the learning rate to $5e-3$, the batch size to 128 and beam search decoding in inference with beam size 10. Early stopping and cross entropy loss are applied during the SC-LSTM and *CSC-LSTM* training. The responses in SC-LSTM and *CSC-LSTM* are delexicalised text where the slot values are replaced by its corresponding slot tokens.

Context size analysis of *CSC-LSTM*: The pre-trained multi-context ConveRT does not only encode the immediate preceding user utterance but in addition a maximum of 10 previous dialogue sentences, i.e., 5 user utterances and 5 system responses (5u5s). To analyse the effect of this context on the performance of the *CSC-LSTM*, multiple models with different context sizes have been trained. And we plot the trend and show exact values of all metric scores in Figure 2 and Table 1 respectively. The 0u0s in Figure 2 and Table 1 means only immediate preceding user utterance without extra context is taken into account for contextual generator *CSC-LSTM* training.

4.2 Contextually and Semantically Conditioned GPT: CSC-GPT

In addition to the contextual generator trained from scratch in Section 4.1, we also explore contextual generation adapted from a pre-trained LM model in this section.

context size	<i>0u0s</i>	<i>1u1s</i>	<i>2u2s</i>	<i>3u3s</i>	<i>4u4s</i>	<i>5u5s</i>
BLEU-4 (%)	29.59	29.76	29.68	29.50	29.46	29.79
Meteor (%)	51.29	51.11	51.21	<u>50.80</u>	50.92	51.22
BERTScore F1 (%)	59.17	59.24	<u>59.02</u>	59.14	59.06	59.13
Variation size	2.05	2.07	2.01	<u>1.99</u>	2.06	2.11

Table 1: The results for BLEU, Meteor, BERTScore and Variation size of *CSC-LSTM* model with variant context size (best results are marked with bold font and worst results are marked with underline) show that the best models exist in *0u0s*, *1u1s* and *5u5s*, while the worst models exist in *2u2s*, *3u3s* and *4u4s*.

We train *CSC-GPT*⁶ on the basis of the pre-trained GPT-2 (Radford et al., 2019). It adopts the generic Transformers (Vaswani et al., 2017). Peng et al. (2020b) already proposed the SC-GPT model, which was continuously training the GPT-2 on (DA, system response) pairs. However, no context information was taken into account in SC-GPT for dialogue response generation. In our proposed *CSC-GPT*, we leverage the extra user context beyond semantic information to guide the generation process. This means, (user utterance, DA, system response) MultiWOZ triplets are continuously trained on the pre-trained GPT-2 model for contextual generation. Given the extra context size analysis result of *CSC-LSTM* model (shown in 5.1) and GPU memory limitation for training the pre-trained LM, only the immediate preceding user utterance is used as context information in the *CSC-GPT*⁷.

In this work, we tackle the generation problem using conditional LM. Given the dialogue dataset $\mathcal{D} = \{(u_n, d_n, r_n)\}_{n=1}^N$ with N samples, the goal is to build a statistical model parameterized by θ to characterize $p_\theta(r|u, d)$, which can be written as the product of a series of conditional probabilities.

$$p_\theta(r|u, d) = \prod_{t=1}^T p_\theta(r_t | r_{<t}, u, d) \quad (2)$$

where $r_{<t}$ indicates all tokens before t . The u represents user utterance; d means the system DA and r is the system response which includes $(r_1, r_2, \dots, r_t, \dots)$ tokens with length T .

Experiment details of *CSC-GPT*: In order to achieve a robust performance comparison, two datasets, namely MultiWOZ 2.1 and MultiWOZ 2.2, are used in SC-GPT and *CSC-GPT*. During

⁶The architecture of *CSC-GPT* is shown in Figure 4 in Appendix

⁷In the Appendix, we also conduct the *CSC-GPT* with different context size: *CSC-GPT (0u0s)* and *CSC-GPT (1u1s)*. The performance comparison in Table 4 supports the extra context size analysis in *CSC-LSTM*: longer dialogue context can not linearly improve the generation performance.

training, the batch size is 16, the maximal epoch is 10, the learning rate is $5e-5$ and early stopping is used. During decoding, we use the top-k (Fan et al., 2018) and nucleus sampling (top-p) (Holtzman et al., 2019) decoding algorithms with top-k equal to 5 and top-p equal to 0.9. This means, the next token distribution is filtered to keep maximal top 5 tokens with highest probability and the cumulative probability above a 0.9 threshold. Due to the computational expense of running large SC-GPT and *CSC-GPT* model, only the top 5 responses are generated.

4.3 BERT Re-ranker: *BERT-R*

In this paper, we propose a **BERT** (Devlin et al., 2019) Re-ranker (*BERT-R*) to select the top generation which is more similar to human sentence and better fits to user context. As the generated responses of the *CSC-LSTM* are delexicalized and have less variability compared with the *CSC-GPT*, we only apply the re-ranker to the *CSC-GPT* model. Adapted from pre-trained BERT, the *BERT-R* is continually trained with task-oriented dialogue data and then fine-tuned on a multiple regression task, where the model learns the relationship between user utterance and system response from the various regression targets defined by multiple metrics scores.

There are two steps in our proposed *BERT-R*: masked LM pre-training and multiple regression fine-tuning. And the BERT-base-uncased⁸ model with 12 layers, 768 hidden units and 12 heads is used in this work.

Masked LM Pre-training The original BERT was pre-trained with the BooksCorpus (Zhu et al., 2015) and English Wikipedia. In order to better generalize to task-oriented dialogues, we continually train the BERT model with a dialogue dataset: DSTC8 (Rastogi et al., 2020), which is a schema-guided dialogue dataset and consists of over 20k

⁸<https://huggingface.co/bert-base-uncased>

annotated multi-domain, task-oriented conversations between a human and a virtual assistant. And Sellam et al. (2020) and Peng et al. (2020b) both continually trained with task-specific data based on the pre-trained LMs for better generalisation. Similar to the masked LM training of the original BERT, only 15% tokens are randomly masked for prediction with cross entropy loss.

Multiple Regression Fine-tuning The ideal generated response should be close to human communication and relevant to preceding user utterance at the same time. The general idea for *BERT-R* is straightforward: using multiple widely used metric scores to guide the model to learn the relationship of user utterance and system response by a multiple regression task, where the regression targets are those metric scores. Those multiple metric scores define how good the generation is from different perspectives to avoid dominance by one single score. Hence, we employ two n -gram matching metrics: BLEU-4 and Meteor score, and a machine learned score: BERTScore, to define how similar system response is with respect to gold reference; and ConveRT cosine similarity to define how contextual system response is with respect to user utterance.

In this work, we continually train *BERT-R* with the guidance of multiple metric scores. Define the user utterance $u = (u_1, \dots, u_l)$ of length l where each u_i is a token and system response $r = (r_1, \dots, r_m)$ of length m . Let $\mathcal{D}' = \{(u_n, r_n, y_n^\tau)\}_{n=1}^{N'}$ be a training dataset of size N' , where τ is a target signal. y^τ represents all metric scores: $y^\tau = (y^{\tau\text{BLEU-4}}, y^{\tau\text{Meteor}}, y^{\tau\text{BERTpre}}, y^{\tau\text{BERTrec}}, y^{\tau\text{BERTfl}}, y^{\tau\text{ConveRT-cs}})$. Hence, the size of τ is 6, which means 6 specific regression layers will be added to the output of [CLS] token in *BERT-R*. y^τ will guide *BERT-R* to learn how similar system response r is with respect to gold reference and how contextual system response r is with respect to user utterance u during the fine-tuning. Given the training data, the goal of fine-tuning is to learn a multiple regression function $f : (u, r) \rightarrow y^\tau$ that predicts different metric scores.

Given the sentence pair (u, r) , the pre-trained *BERT-R* returns a sequence of contextualized vectors:

$$v_{[\text{CLS}]}, v_{u_1}, \dots, v_{u_l}, v_{r_1}, \dots, v_{r_m} = \text{BERT-R}(u, r) \quad (3)$$

where $v_{[\text{CLS}]}$ is the *BERT-R* output representation for the special [CLS] token, which can be further fine-tuned for classification or regression task. As described by Devlin et al. (2019), we add separate linear layers on top of the [CLS] vector to predict different metric scores:

$$\hat{y}^\tau = f(u, r) = \mathbf{W}^\tau v_{[\text{CLS}]} + \mathbf{b}^\tau \quad (4)$$

where \mathbf{W}^τ and \mathbf{b}^τ are the weight matrix and bias vector respectively. And we use the Eq. 5, the sum of all target-specific regression loss to fine-tune *BERT-R*.

$$l_{\text{fine-tuning}} = \sum_{\tau} \left(\frac{1}{N'} \sum_{n=1}^{N'} \|\hat{y}_n^\tau - y_n^\tau\|^2 \right) \quad (5)$$

After fine-tuning, the *BERT-R* is used to select the top generation with the highest score: the sum of all regression output of *BERT-R*, which is shown in Eq. 6:

$$S_{\text{BERT-R}} = \sum_{\tau} \text{BERT-R}(u, r) \quad (6)$$

Experiment details of *BERT-R*: During masked LM pre-training, batch size is 32, maximal epoch is 10 and learning rate is $5e-5$. And early stopping is used to avoid over-fitting on the DSTC8 training dataset.

For the fine-tuning of *BERT-R*, we generate the top 5 responses for MultiWOZ data with *CSC-GPT* model firstly. And in order to clean the system generated responses, we have the following procedures:

- 1) The duplicated system responses are removed.
- 2) The last turn of every dialogue is removed, where there are always “thank you” and “good bye”, kind of non-informative sentences.
- 3) In order to let the network glance what the human communication looks like, we add the gold reference for the user utterance in the training dataset. In addition, we need to remove the system responses which are same as the gold reference beforehand, to comply with the rule 1).

After finishing the above process, we compute the target score respectively for the label of different regression layers. During the training of the multiple regression task, the batch size is set to 32, learning rate $1e-5$, and early stopping is used to save the best BERT re-ranker.

	MultiWOZ 2.1				MultiWOZ 2.2			
	BLEU-4 (%)	Meteor (%)	BERTScore F1 (%)	Variation size	BLEU-4 (%)	Meteor (%)	BERTScore F1 (%)	Variation size
SC-LSTM (Wen et al., 2015)	28.76	49.93	58.71	1.00/1.00	-	-	-	-
<i>CSC-LSTM (5u5s)</i>	29.79	51.22	59.13	1.35/2.11	-	-	-	-
SC-GPT (Peng et al., 2020b)	28.95	50.22	91.96	2.27/6.56	28.53	49.80	91.95	2.78/6.70
<i>CSC-GPT</i>	29.91	51.34	92.08	2.29/6.66	29.41	51.10	92.08	2.81/6.82
<i>BERT-R</i>	32.37	54.01	92.40	2.22/6.34	31.68	53.65	92.39	2.81/6.82

Table 2: The results for BLEU, Meteor, BERTScore and variation size of top one generation in all models demonstrate that our proposed contextual models in: *CSC-LSTM* against SC-LSTM and *CSC-GPT* against SC-GPT, both outperform the corresponding baseline. Meanwhile, the proposed re-ranker *BERT-R* also highly improve the metric scores compared with all other models.

5 The Experiment Results

The experimental results of all models introduced in Section 4 are presented in this section. To ensure a consistent performance comparison, we compute the metric scores based on the top one generation of all models. Table 2 shows the results of all automatic metrics.

Furthermore, a **human evaluation** has been conducted. We randomly sampled 100 dialogues and their corresponding top one generations from our proposed models as well as the baselines. We recruited three annotators with relevant background in SDS to evaluate the responses generated by different models. Each rater was presented the complete preceding dialogue and asked to rate if “The highlighted system response could plausibly have been produced by a human” (natural) and if “The highlighted system response fits well to the previous dialog” (contextual). Each metric is rated on a 5-point Likert scale, where 1 is “not agree at all”, 5 is “fully agree”. In order to guarantee the strictness of human evaluation, the human judges have no information about the origin of the utterances, i.e., which model generated the utterance. Table 3 shows the human evaluation results.

	Contextual	Natural
SC-LSTM (Wen et al., 2015)	3.96	4.04
<i>CSC-LSTM (5u5s)</i>	4.21*	4.16*
SC-GPT (Peng et al., 2020b)	4.00	4.14
<i>CSC-GPT</i>	4.25+	4.27+
<i>BERT-R</i>	4.18	4.26

Table 3: The results of human evaluation on natural and contextual score of all models. (*: p -value < 0.01, comparison with SC-LSTM baseline; +: p -value < 0.1, comparison with SC-GPT baseline) show the superiority of our proposed contextual models.

5.1 Experiment results of *CSC-LSTM*

The automatic metric scores comparing SC-LSTM and *CSC-LSTM* in Table 2 show that *CSC-LSTM* has the overall better performance in BLEU-4, Meteor, BERTScore and variation size compared to the baseline. The variation size results show *CSC-LSTM* can generate more variant responses per DA, which may indicate a more contextual fitting response, while the SC-LSTM only generate the same utterances each time. The performance comparison between SC-LSTM and *CSC-LSTM* in Table 2 support our initial assumption that context helps to generate good system utterances. Especially the increase in variation size is of importance as it indicates that the resulting utterances of *CSC-LSTM* indeed be different for different contexts.

This has been validated by the human evaluation of SC-LSTM and *CSC-LSTM* in Table 3. It shows that the variation introduced by *CSC-LSTM* actually results in utterances that fit significantly better to the preceding dialogue and are perceived as significantly more natural. Overall, the performance comparison between SC-LSTM and *CSC-LSTM* on automatic metrics and human evaluation demonstrate the dialogue history contributes to contextual and variant responses.

The context size analysis of *CSC-LSTM* demonstrate that the automatic metric scores are influenced by the length of the context. All metrics show similar curves over the different contextual model in Figure 2. The both ends of the curves have better performance than the inner part. The Table 1 shows the best BLEU-4 and variation size are both achieved for context sizes of *5u5s*; while *0u0s* has the best Meteor and *1u1s* has the best BERTScore. Both show that the contextual models *0u0s*, *1u1s* and *5u5s* generally outperform the

2u2s, 3u3s and 4u4s. Hence, the investigation of the impact of context size for *CSC-LSTM* generation indicates that longer context does not linearly result in better performance, which is further confirmed in the performance comparison between *CSC-GPT(0u0s)* and *CSC-GPT(1u1s)* shown in Table 4 in Appendix. Evidently, all contextual models achieve better performance than the baseline (SC-LSTM in Table 2). We therefore conclude that the immediate preceding user utterance yields the indispensable information for contextual generation. With limited memory, using only immediate preceding user utterance without extra context can be regarded as a balanced option that we directly apply for training the *CSC-GPT*.

5.2 Experiment results of *CSC-GPT*

All scores between SC-GPT and *CSC-GPT* in Table 2 demonstrate that the *CSC-GPT* is superior to the baseline SC-GPT for both datasets, MultiWOZ 2.1 and MultiWOZ 2.2. Our assumption is again confirmed in GPT-based generation model: dialogue history contributes to contextual and variant response, even though only one preceding user utterance is taken into account.

When comparing the human evaluation results of SC-GPT and *CSC-GPT* in Table 3, the assumption is further supported: adding context to the generation process results in more natural and contextual responses. The *CSC-GPT* generator even achieves the best rating both in terms of natural and contextual score compared to all other models. This means that enhancing an already powerful pre-trained model with context is essential for its application within dialogue systems.

5.3 Experiment results of *BERT-R*

Our proposed *BERT-R* selects the top one response from 5 *CSC-GPT* generations with the highest score in Eq. 6. The metric scores of *CSC-GPT* and *BERT-R* in Table 2 show that the selected generation by *BERT-R* has a significant improvement on BLEU-4, Meteor and BERTScore⁹, with a little loss on variation size compared to the top one generation in *CSC-GPT*. This is in line with human evaluation results where the *CSC-GPT* achieves slightly better scores both in naturalness and contextualness than *BERT-R*, even though *BERT-R* clearly shows better results compared to the SC-GPT baseline.

⁹also improvement on ConveRT cosine similarity, which is shown in the Appendix.

6 Conclusion and Future Work

In this paper, we propose two contextual generation models: *CSC-LSTM* trained from scratch and *CSC-GPT* adapted from pre-trained GPT-2. Both integrate dialogue context information into NLG for generating more variant and contextual response in task-oriented dialogue systems.

In the experiment of *CSC-LSTM* against SC-LSTM and *CSC-GPT* against SC-GPT, our proposed contextual models both improve the generation performance in automatic metrics, thus showing that *CSC-LSTM* and *CSC-GPT* are able to capture better the contextual needs resulting in a higher similarity to the data. This is further underpinned by the number of variations. More variant responses are generated per DA in *CSC-LSTM*, while the SC-LSTM only generates the same utterances each time. Furthermore, the variation size of GPT-based generators is higher than SC-LSTM based models. The possible reason is the pre-trained GPT-2 contributes to more diverse responses by default. The human evaluation results in Table 3 not only demonstrate the contextual model can generate more contextual and natural response compared with their baseline respectively, but also show GPT-2 contextual model *CSC-GPT* is superior than *CSC-LSTM*.

An investigation of the impact of context size for dialogue response generation in *CSC-LSTM* indicates that longer context does not automatically result in better performance. However, all variant *CSC-LSTM* models have better performance than baseline, which means the immediate preceding user utterance contains the most contextual information for generation. This is also verified in GPT-2 contextual generators, even only immediate preceding user utterance is taken into account, the *CSC-GPT* model outperforms SC-GPT both on automatic metrics and human evaluation.

In addition to the above mentioned two contextual models, we also present a re-ranker for *CSC-GPT* contextual model. Adapted from pre-trained BERT, the *BERT-R* continually train on multi-domain dialogues and fine-tune on a multiple regression task to learn the relationship between user context and system response by the metric guidance of BLEU-4, Meteor, BERTScore and ConveRT cosine similarity. Finally, the top one generation selected by *BERT-R* has significant superiority in BLEU-4, Meteor, BERTScore and ConveRT cosine similarity compared with top one

generation in *CSC-GPT*. This means, that our proposed *BERT-R* works from the guidance of metric scores and can choose the generation with highest score. However, *CSC-GPT* slightly outperforms *BERT-R* in variation size and human evaluation. The possible reason is that the existing automatic metrics still have bias with human judgments (Chaganty et al., 2018).

In the future, we will further explore the performance of *BERT-R* with the guidance of other automatic metrics, which have higher correlation with human judgements. Furthermore, there seems to be a link between the variation size metric and the human evaluation scores, which will also be part of future work.

References

- 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.
- Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*.
- Arun Tejasvi Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evaluation. In *ACL (1)*.
- Zhiyu Chen, Harini Eavani, Wenhua Chen, Yinyin Liu, and William Yang Wang. 2020. Few-shot nlg with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190.
- 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.
- Ondřej Dušek and Filip Jurcicek. 2016. A context-aware natural language generator for dialogue systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 185–190.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 422–428.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *ACL (1)*.
- Matthew Henderson, Paweł Budzianowski, Iñigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrkšić, Georgios Spithourakis, Pei-Hao Su, Ivan Vulić, et al. 2019. A repository of conversational datasets. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 1–10.
- Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. Convert: Efficient and accurate conversational representations from transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2161–2174.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.
- Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6505–6520.
- Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- 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.
- Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayan-deh, Lars Liden, and Jianfeng Gao. 2020a. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *arXiv preprint arXiv:2005.05298*.
- Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujuan Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020b. Few-shot natural language generation for task-oriented dialog. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 172–182.

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

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

David Reitter, Frank Keller, and Johanna D Moore. 2006. Computational modelling of structural priming in dialogue. In *Proceedings of the human language technology conference of the naacl, companion volume: Short papers*, pages 121–124.

Chinnadhurai Sankar, Sandeep Subramanian, Chris Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. In *ACL (1)*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892.

Bo-Hsiang Tseng, Florian Kreyszig, Paweł Budzianowski, Iñigo Casanueva, Yen-Chen Wu, Stefan Ultes, and Milica Gasic. 2018. Variational cross-domain natural language generation for spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 338–343.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

TH Wen, M Gašić, N Mrkšić, PH Su, D Vandyke, and S Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Conference Proceedings-EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.

Xiaoxue Zang, Abhinav Rastogi, and Jindong Chen. 2020. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies

and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Appendices

In order to better understand what our proposed architectures look like, please find the following Figure 3 and Figure 4. The model *CSC-GPT* generates contextual response with the guidance of immediate preceding user utterance and DA. In the *CSC-LSTM*, the SC-LSTM was proposed in Wen et al. (2015) and an additional cell was introduced into the LSTM cell to gate the DA information. The original LSTM cell follows:

$$\begin{aligned}
 i_t &= \sigma(\mathbf{W}_{wi}w_t + \mathbf{W}_{hi}h_{t-1}) \\
 f_t &= \sigma(\mathbf{W}_{wf}w_t + \mathbf{W}_{hf}h_{t-1}) \\
 o_t &= \sigma(\mathbf{W}_{wo}w_t + \mathbf{W}_{ho}h_{t-1}) \\
 \hat{c}_t &= \tanh(\mathbf{W}_{wc}w_t + \mathbf{W}_{hc}h_{t-1}) \\
 c_t &= f_t \odot c_{t-1} + i_t \odot \hat{c}_t \\
 h_t &= o_t \odot \tanh(c_t)
 \end{aligned} \tag{7}$$

In SC-LSTM, the d_0 starts from an one-hot vector, at each time step the DA cell decides what information should be retained for future time steps and discards the others. Like:

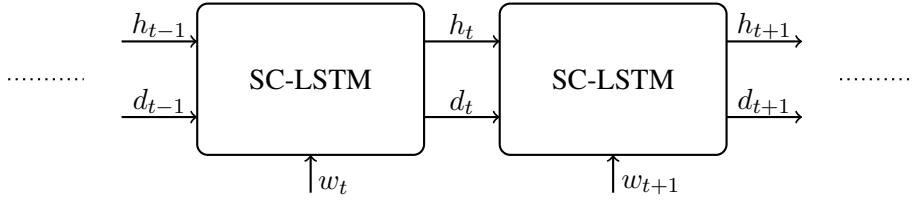
$$\begin{aligned}
 r_t &= \sigma(\mathbf{W}_{wr}w_t + \alpha(\mathbf{W}_{hr}h_{t-1})) \\
 d_t &= r_t \cdot d_{t-1}
 \end{aligned} \tag{8}$$

Then, the value cell in Eq. 7 also depends on the DA,

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t + \tanh(\mathbf{W}_{dc}d_t) \tag{9}$$

Finally, the hidden state is further updated by new value cell. In our proposed *CSC-LSTM*, not only DA cell is added, but the SC-LSTM cell is initialized by contextual ConveRT embedding.

Given the pre-trained model can handle the longer dialogue context for generation, so we also trained *CSC-GPT* contextual model with one more turn context besides the immediate preceding user utterance, i.e. *1u1s*. And the results comparison between *CSC-GPT (0u0s)* and *CSC-GPT (1u1s)* is shown in Table 4. We can know that the BLEU-4 and Meteor of *CSC-GPT (0u0s)* outperforms *CSC-GPT (1u1s)*, meanwhile, the BERTScore and variation size have no big difference. This further demonstrate that longer context can not directly result in better performance. The maximal length of input in *CSC-GPT (0u0s)* is 120, however, *CSC-GPT (1u1s)* is 190 with more GPU memory. Hence,



h_0 refers to Eq. 1, dialogue context is encoded by ConveRT and projected to decoder space with the Eq. 1
 d_0 is DA one-hot embedding, like $(0, 0, \dots, 1, \dots)$ represents “hotel{ request(area=?)}”
 w_t is the input token at time slot t

Figure 3: The architecture of *CSC-LSTM*.

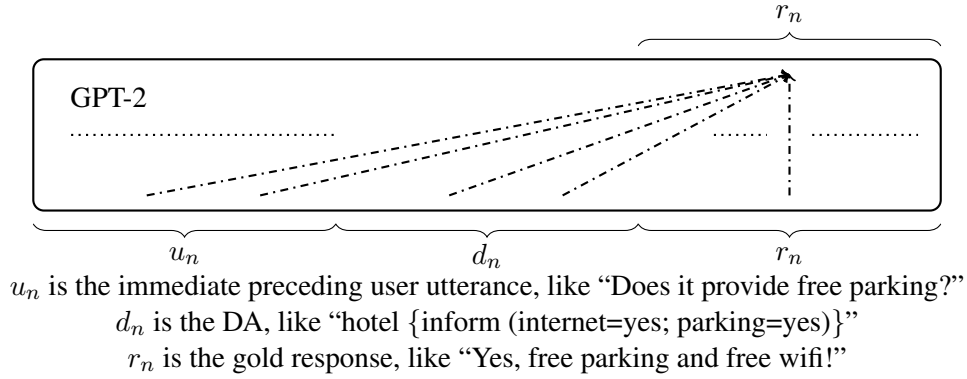


Figure 4: The architecture of *CSC-GPT*. The immediate preceding user utterance and MR together guide the contextual response generation.

we recommend only immediate preceding user utterance is taken into account for contextual generation in task-oriented dialogue system.

The *BERT-R* is trained with multiple metric scores: BLEU-4, Meteor, BERTScore and ConveRT cosine similarity. In order to make consistent comparison of all models, we don’t show the ConveRT cosine similarity of *BERT-R* in the main paper. The Table 5 shows that all scores get improvement.

There are several use cases in Table 6. And by system response comparison of SC-LSTM against *CSC-LSTM* and SC-GPT against *CSC-GPT* given preceding dialogue context, the Table 6 shows the importance of dialogue context for natural and contextual response.

	MultiWOZ 2.1				MultiWOZ 2.2			
	BLEU-4 (%)	Meteor (%)	BERTScore F1 (%)	Variation size	BLEU-4 (%)	Meteor (%)	BERTScore F1 (%)	Variation size
<i>CSC-GPT (0u0s)</i>	29.91	51.34	92.08	2.29/6.66	29.41	51.10	92.08	2.81/6.82
<i>CSC-GPT (1u1s)</i>	29.70	51.02	92.08	2.29/6.67	28.80	50.32	91.94	2.81/6.81

Table 4: The results for BLEU, Meteor, BERTScore and Variation size of top one *CSC-GPT (0u0s)* generation and *CSC-GPT (1u1s)* show that *CSC-GPT (0u0s)* outperforms *CSC-GPT (1u1s)* and support the context size analysis of *CSC-LSTM*: the longer context in contextual model can not linearly result in better performance.

	MultiWOZ 2.1				MultiWOZ 2.2			
	BLEU-4 (%)	Meteor (%)	BERTScore F1 (%)	ConveRT cs	BLEU-4 (%)	Meteor (%)	BERTScore F1 (%)	ConveRT cs
<i>CSC-GPT</i>	29.91	51.34	92.08	33.41	29.41	51.10	92.08	33.81
<i>BERT-R</i>	32.37	54.01	92.40	34.70	31.68	53.65	92.39	35.00

Table 5: The results for BLEU, Meteor, BERTScore and ConveRT cosine similarity of top one *CSC-GPT* generation and the selected one by *BERT-R* demonstrate that all target metrics used in *BERT-R* have a significant improvement through re-ranking.

context	User: Hi, I'm looking for a hotel called aylesbray lodge guest house. Do you have anything with that name?
system response	<p>SC-LSTM: It is a moderately priced hotel in the south. Would you like me to book it for you?</p> <p>CSC-LSTM: <u>Yes</u>, it is a hotel in the south in the moderately priced price range. Would you like to book a room?</p> <p>SC-GPT: It is a moderately priced hotel in the south area. Would you like me to book it for you?</p> <p>CSC-GPT: <u>I do</u>, it is a moderately priced hotel in the south. Would you like to make a booking?</p>
context	<p>User: I'm looking for a nightclub to go to. Are there any in the centre of town?</p> <p>System: There are 5 nightclubs in the centre. Ballare is one of them. Would that work for you?</p> <p>User: Is there an entrance fee for Ballare?</p>
system response	<p>SC-LSTM: The entrance fee is 5 pounds.</p> <p>CSC-LSTM <u>Yes</u>, it is 5 pounds .</p> <p>SC-GPT: The entrance fee is 5 pounds.</p> <p>CSC-GPT: <u>Yes</u>, it is 5 pounds <u>to get in</u>.</p>
context	User: Hi, I am trying to plan my trip and could use some help with a particular restaurant.
system response	<p>SC-LSTM: Do you have a preference for area or price range?</p> <p>CSC-LSTM: Do you have a certain area or price range <u>in mind</u>?</p> <p>SC-GPT: <u>Sure</u>, what kind of food are you looking for and what area and price range?</p> <p>CSC-GPT: <u>I would love to assist you!</u> What type of food are you looking for and what price range?</p>
context	<p>User: Can you also help me find a train to Bishops Stortford?</p> <p>System: There are more than 70 trains to Bishops Stratford. Could you tell me what day you would like to leave, where you would like to leave from, and the times?</p> <p>User: I am leaving Cambridge on Sunday, can you please tell me the travel time on that ride?</p> <p>System: Sure, it is a 38 minute trip. Would you like a ticket?</p> <p>User: I need to leave after 14:45 also.</p>
system response	<p>SC-LSTM: The tr3600 leaves Cambridge at 15:29.</p> <p>CSC-LSTM: Tr3600 leaves Cambridge at 15:29. <u>Would that works for you?</u></p> <p>SC-GPT: Tr3600 leaves Cambridge at 15:29.</p> <p>CSC-GPT: <u>How about tr3600?</u> It leaves Cambridge at 15:29.</p>

Table 6: System response of different models given preceding dialogue context show that the contextual models: *CSC-LSTM* and *CSC-GPT*, can generate more natural responses that fit better to dialogue history.