

# AuGPT: Auxiliary Tasks and Data Augmentation for End-To-End Dialogue with Pre-Trained Language Models

Jonáš Kulhánek,<sup>1,2,3</sup> Vojtěch Hudeček,<sup>1</sup> Tomáš Nekvinda<sup>1</sup> and Ondřej Dušek<sup>1</sup>

<sup>1</sup>Charles University, Faculty of Mathematics and Physics, Prague, Czechia

<sup>2</sup>Czech Technical University in Prague, Czech Institute of Informatics, Robotics and Cybernetics

<sup>3</sup>Czech Technical University in Prague, Faculty of Electrical Engineering

jonas.kulhanek@cvut.cz, {hudecek,nekvinda,odusek}@ufal.mff.cuni.cz

## Abstract

Attention-based pre-trained language models such as GPT-2 brought considerable progress to end-to-end dialogue modelling. However, they also present considerable risks for task-oriented dialogue, such as lack of knowledge grounding or diversity. To address these issues, we introduce modified training objectives for language model finetuning, and we employ massive data augmentation via back-translation to increase the diversity of the training data. We further examine the possibilities of combining data from multiples sources to improve performance on the target dataset. We carefully evaluate our contributions with both human and automatic methods. Our model substantially outperforms the baseline on the MultiWOZ data and shows competitive performance with state of the art in both automatic and human evaluation.

## 1 Introduction

Unlike traditional task-oriented systems based on modularized pipelines (Young et al., 2013; Gao et al., 2019), end-to-end dialogue systems integrate nearly all functionality required to hold a dialogue into a single neural network (Wen et al., 2017; Eric et al., 2017; Lei et al., 2018), reducing error-propagation and data annotation requirements. While these systems are not yet ready for production use, they made considerable progress in recent years, especially with the advent of pre-trained neural language models (LMs) (Devlin et al., 2019; Radford et al., 2019; Zhang et al., 2020c). Systems such as GPT-2 finetuned by Budzianowski and Vulić (2019) show that with an LM pre-trained on a large number of general-domain dialogues without annotation, only small amounts of data are required to perform well in a given task-oriented domain.

On the other hand, the pre-trained LMs run enormous risks. First, solely training for response generation may result in a lack of grounding for

the responses, where the LM hallucinates words without any relation to the database. This has been addressed by multi-task training and auxiliary training objectives (Peng et al., 2021) to an extent. Second, finetuning on small datasets may reduce response diversity and fluency due to neural networks’ known propensity for catastrophic forgetting (Greco et al., 2019) – the model overfits the finetuning dataset too tightly, “forgetting” the pre-trained language modeling capabilities.

This paper presents an end-to-end model for multi-domain task-oriented response generation on the MultiWOZ data (Budzianowski et al., 2018),<sup>1</sup> where we address the above problems with pre-trained LMs. AuGPT is based on the GPT-2 LM and Peng et al. (2021)’s basic approach. Our contributions can be summarized as follows:

- We introduce a new dialogue consistency classification task based on subtle changes to the dialogue state (instead of fully random resampling) used as an auxiliary training objective, and we demonstrate its performance improvements.
- We present a novel application of token unlikelihood loss (Welleck et al., 2020) in task-oriented dialogue to further improve diversity of our model’s responses.
- We apply pre-training on additional datasets and massive data augmentation using back-translation via multiple languages (Sennrich et al., 2016) and demonstrate that both markedly improve task-oriented dialogue performance.
- We compare our model to multiple baselines on MultiWOZ in a corpus-based and simulated evaluation. We also include human evaluation results from a shared task competition, as well as detailed manual error analysis.

We publish our augmented training data, source code, and pre-trained models on GitHub.<sup>2</sup>

<sup>1</sup><https://convlab.github.io>

<sup>2</sup><https://github.com/ufal/augpt>

## 2 Related Work

While the first attempts to build generative end-to-end task-oriented systems mimicked the traditional dialogue system components (Wen et al., 2017), the task was soon recast as a sequence prediction problem in a two-stage setup. A sequence-to-sequence (seq2seq) model first generates the belief state based on dialogue context, then generates the system response based on the context and the belief state (Sequicity; Lei et al., 2018).

Recently, large-scale multi-domain task-oriented datasets were proposed (Budzianowski et al., 2018; Byrne et al., 2019; Rastogi et al., 2020). To address multiple domains, Zhang et al. (2020a) introduce the LABES-S2S model that – in addition to a two-stage seq2seq approach – models belief states as discrete latent variables. Zhang et al. (2020b) present DAMD, a three-stage seq2seq architecture which explicitly decodes the system action. They optimize for multiple good actions given a single belief state. Qin et al. (2020) investigate sharing of domain knowledge and performance on unseen domains. Lubis et al. (2020)’s LAVA model employs reinforcement learning over latent system actions initialized using a variational autoencoder.

The line of research closest to our work makes use of large pre-trained LMs based on the transformer architecture (Vaswani et al., 2017) such as GPT-2 (Radford et al., 2019) or BERT (Devlin et al., 2019). For example, Wu et al. (2020) propose finetuning BERT (Devlin et al., 2019) for task-oriented dialogue, Zhang et al. (2020c) extended the GPT-2 LM to model open-domain chat-chat.

We follow research initiated by Budzianowski and Vulić (2019), who use GPT-2 to model multi-domain task-oriented dialogues. Recently, three similar modifications to their model were proposed, namely SOLOIST (Peng et al., 2021), SimpleTOD (Hosseini-Asl et al., 2020), and the approach by Ham et al. (2020). Our work extends these models and proposes a novel training approach and data augmentation strategies based on back-translation (Edunov et al., 2018; Federmann et al., 2019). Earlier works used a single pivot language (Jin et al., 2018; Einolghozati et al., 2019), whereas our work applies 10 languages to increase variability.

## 3 Method

The task-oriented setting requires the dialogue system to respond adequately to the user’s input and fulfill its goal, e.g., booking a train or request-

ing restaurant details. The system must process the user’s input, keep track of the belief state (user preferences regarding individual slots, i.e., in-domain attributes) and generate a relevant response in natural language. It must also interact with a database to incorporate external information into its responses (see Figure 1 for an example). Following Budzianowski and Vulić (2019), we choose the GPT-2 LM as our backbone and use the LM to model both the belief state and the response.

### 3.1 Model Representation

The training instances for an LM-based task-oriented dialogue system can be considered as tuples  $(c, b, d, r)$ , where  $c$  is the context (i.e., a concatenation of all previous utterances in the dialogue – both system’s and user’s),  $b$  is the system’s belief state (used to query the database),  $d$  are the database results, and  $r$  is the system’s response.

In our case, the dialogue system handles multiple domains and the belief state is a set of pairs (*domain name*, *domain belief*), where the *domain belief* is an assignment of values into slots, i.e., a set of pairs (*slot name*, *value*) (see Example 1). Similarly, the database results  $d$  are a set of pairs (*domain name*, *domain database results*), where the *domain database results* are an ordered list of entities returned by the database. We further define the *database result counts*  $d_c$  denoting the number of results in  $d$  for each domain.

Ideally, we would like our system to model the probability distribution over possible responses conditioned on the context  $p(r|c)$ . To simplify computation and model external database queries, we factorize this distribution as follows:

$$\begin{aligned} p(r|c) &= \sum_d p(r|d, c) p(d|c) \\ &= \sum_d \sum_b p(r|d, b, c) p(d|b) p(b|c) \quad (1) \\ &= \sum_b p(r|Query(b), b, c) p(b|c), \end{aligned}$$

where  $p(d|b)$  is a deterministic distribution over the database results, and *Query* is a function returning database results.

Using this factorization allows the model to process the context, query the database and generate a response based on database results. However, generating responses directly would result in data sparsity issues with rare tokens (e.g., venue names or reference numbers). To maximally reuse the training samples, we choose to train our model

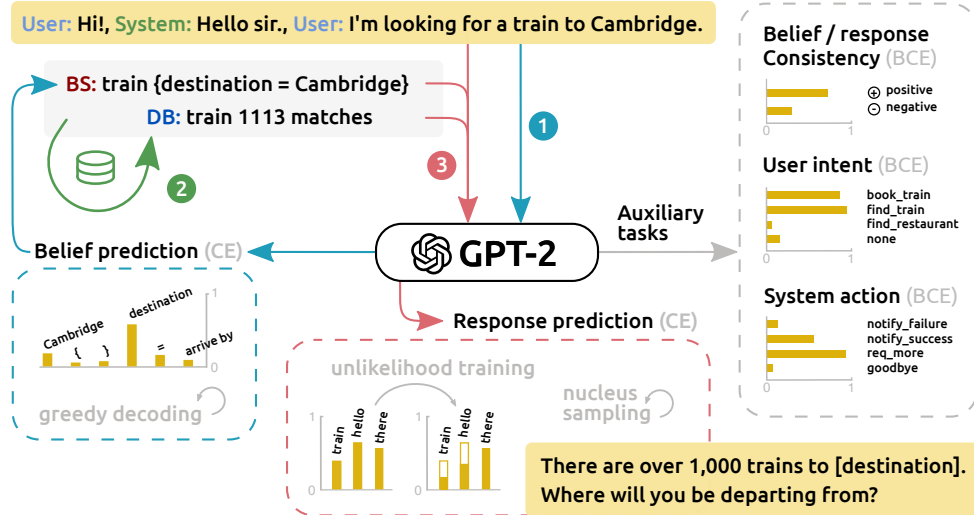


Figure 1: The architecture of AuGPT. The pipeline runs in two stages. First, a finetuned GPT-2 LM is used to predict a belief. Then the database results are obtained and everything is passed to the GPT-2 again to predict a final delexicalized response, along with possible auxiliary tasks (belief consistency, intent classification, system action classification). Unlikelihood loss is used for response prediction training.

on *delexicalized responses* denoted  $\bar{r}$ , where slot values are replaced with placeholders (Wen et al., 2015). During inference, the responses are lexicalized back deterministically using the belief state and the database results. We assume perfect lexicalization, i.e., always being able to lexicalize the response  $\bar{r}$  back based on  $d$  and  $b$ .<sup>3</sup>

Both the database lookup and the lexicalization are deterministic, and the delexicalized response  $\bar{r}$  does not depend on the database results  $d$ , but only on their counts  $d_c$ . Therefore, the distribution  $p(r|d, b, c)$  is equal to the distribution  $p(\bar{r}|d_c, b, c)$ , and by maximizing its likelihood we are achieving the goal of maximizing the likelihood of  $p(r|c)$ .

We use the same language model  $\hat{p}$  to model the belief state and to generate the delexicalized prediction. That is,

$$p(\bar{r}|d_c, b, c) \approx \hat{p}(\bar{r}|d_c, b, c, \theta) \quad (2)$$

$$p(b|c) \approx \hat{p}(b|\emptyset, \emptyset, c, \theta), \quad (3)$$

where we denote the model’s parameters as  $\theta$ .

In the MultiWOZ dataset (Budzianowski et al., 2018; Eric et al., 2020, see Section 4), responses are delexicalized by replacing concrete values with placeholder tokens of the form *domain\_slot*. For better generalization across domains, we chose to only use *slot* instead as responses rarely involve more than one domain. We train our model to predict the *active domain* by outputting it first in the

<sup>3</sup>We found in our experiments on the MultiWOZ data (see Section 4) that this assumption was almost always fulfilled.

belief state (remaining domains follow in lexicographical order). The predicted active domain is then used during lexicalization.<sup>4</sup>

Belief state: train { leave at=15:30,  
arrive by=17:15 },  
hotel { price range = cheap }  
DB: train 23 matches, hotel no match

Example 1: String format for AuGPT’s belief state and database result count.

To fully exploit natural language pre-training of our LM, we represent the belief state and database result counts as strings containing as few special tokens as possible (see Example 1).

### 3.2 Model Training

Although parameters are shared for the belief state predictor and the delexicalized response predictor, the training objectives differ slightly. We use cross-entropy loss for both; response prediction uses unlikelihood loss (Welleck et al., 2020; Li et al., 2020) as an additional objective. Unlikelihood loss penalizes repeated tokens, which helps the model avoid repetitions and increases output diversity.

To help the model learn a better internal representation from the data, we employ additional auxiliary tasks. Similarly to Devlin et al. (2019)

<sup>4</sup>A disadvantage of this approach is that we cannot determine the active domain if the belief state is empty. However, in such a case the lexicalization would fail anyway, so the system’s performance is not affected by this decision.

and Peng et al. (2021), we train a binary classifier to detect dialogue inconsistencies. In each training batch, we corrupt half of the samples by randomly applying one or more of the following changes with the same probability:

1. We replace the belief state  $b$  with another belief state, sampled uniformly randomly from the training data.
2. We replace the delexicalized response  $\bar{r}$  with a different randomly chosen one. If this change is applied in combination with the first one, the delexicalized response and the belief state are taken from the same random sample.
3. A different valid value is uniformly sampled for each slot in the belief state. In this case, the domain names and domain order are unchanged (i.e., the active domain is the same).

The first two changes are identical to Peng et al. (2021). The third one is a new one which we find very useful – it is much more challenging to detect if the belief state was changed when the domain stays the same. Consistency detection employs an affine binary classifier on top of last response token logits, trained using binary cross-entropy (BCE).

We also experiment with additional two classifiers predicting the user intent and the system action. These are implemented as two fully-connected layers attached to the last context token and the last database result token logits, respectively. However, based on our experimental results (see Table 4), we decided not to use these tasks in the final model.

We train the whole pipeline by optimizing the non-weighted sum of individual component losses, i.e., cross-entropy for belief state and response prediction, unlikelihood loss for the response, and BCE for consistency detection.

### 3.3 Response Generation

For each user input, the system goes through several stages (see Figure 1): (1) Previous dialogue context is passed to the LM, which greedily generates the string representation of the belief state. (2) The belief state is parsed and passed to the database handler. (3) The database handler returns a set of results for each domain. (4) A string representation of database result counts is created (see Example 1). (5) The context, belief state and database results are concatenated and passed again to the LM. We use nucleus sampling (Holtzman et al., 2020) to gener-

ate the delexicalized response.<sup>5</sup> (6) Placeholders in the delexicalized response are replaced by values from the database results and the belief state.

### 3.4 Data Augmentation

Following its successful usage in other NLP tasks, (Konstas et al., 2017; Elder et al., 2020), we experiment with data augmentation using paraphrases. In our setup, we generate multiple paraphrases for each training utterance and use them to augment the training data. This way, we effectively increase the variability of the data.

Various data-driven approaches for paraphrasing were proposed, the majority of them corpora-based (Madnani and Dorr, 2010). Recently, machine translation systems showed strong performance in generating paraphrases using back-translation (Sennrich et al., 2016; Edunov et al., 2018; Federmann et al., 2019), i.e., translating an English text into an intermediate language and then translating the result back into English. We use two different Transformer-based machine translation systems to paraphrase our data. We used Edunov et al. (2018)’s system with French and the system of Macháček et al. (2020); Zouhar et al. (2021) with additional 40 pivot languages. Based on empirical analysis of translation quality, we chose 10 pivot languages for our data – we obtain 10 different paraphrases for each input utterance.<sup>6</sup> When training, we choose the input user utterance uniformly at random from the set of all 10+1 variants of the utterance (backtranslation outputs and the original one).

## 4 Experiments

### 4.1 Datasets

As our primary dataset, we use MultiWOZ 2.1, a de-noised version of MultiWOZ 2.0 (Budzianowski et al., 2018). We also used the 2.0 version to compare to previous works. The dataset contains 7 distinct domains (all related to tourist information, e.g., hotels, restaurants) and 10,438 dialogues, 7,032 of which are multi-domain.

We experiment with pre-training our model on additional datasets. For the pre-training phase, we use Taskmaster-1 (Byrne et al., 2019) and Schema-

<sup>5</sup>We found nucleus sampling useful for generating the response since it increases diversity, but we prefer greedy decoding for the belief state with a fixed structure.

<sup>6</sup>Pivot languages used: Albanian, Arabic, Bulgarian, Bosnian, French, German, Russian, Spanish, Slovak, Swedish.

method	MultiWOZ 2.0			MultiWOZ 2.1		
	inform	success	BLEU	inform	success	BLEU
Human	91.0	82.7	–	86.3	79.1	–
<b>AuGPT</b>	83.1	70.1	17.2	83.5	67.3	17.2
SOLOIST (Peng et al., 2021)	85.5	72.9	16.5	–	–	–
SimpleTOD (Hosseini-Asl et al., 2020)	84.4	70.1	15.1	85.0	70.5	15.2
LABES-S2S (Zhang et al., 2020a)	–	–	–	78.1	67.1	18.3
DAMD (Zhang et al., 2020b)	76.3	60.4	16.6	–	–	–
MD-Sequicity (Zhang et al., 2020b)	86.6	71.6	16.8	–	–	–
LAVA (Lubis et al., 2020)	91.8	81.8	12.0	–	–	–

Table 1: Comparison with previous works on the MultiWOZ dataset (see Section 4.4 for a description of the metrics). *MD-Sequicity* is a variant of Lei et al. (2018)’s model, extended for a multi-domain setting.

method	complete	success	book	inform			turn	
				P	R	F1	succ	all
<b>AuGPT</b>	89.4	60.1	85.7	64.5	82.1	70.3	12.7	14.6
DAMD (Zhang et al., 2020b)	39.5	34.3	51.4	60.4	59.8	56.3	15.8	29.8
Sequicity (Lei et al., 2018)	23.1	9.8	4.1	33.0	32.7	29.9	12.2	32.6

Table 2: ConvLab evaluation comparison with other works (see Section 4.5 for a description of the metrics).

Guided Dialogue (Rastogi et al., 2020).<sup>7</sup> Both Taskmaster-1 and Schema-Guided Dialogue are multi-domain, task-oriented, large dialogue corpora consisting of 12,215 and 22,825 dialogues, respectively. Taskmaster-1 was obtained using the Wizard-of-Oz and self-dialogue methods, while the collection of Schema-Guided Dialogue is somewhat artificial – humans are only employed to paraphrase machine-generated utterances.

## 4.2 Data Preprocessing

Although the MultiWOZ 2.1 dataset was collected by humans, it contains a lot of inconsistencies. We hypothesize that when using only *clean* samples which are consistent with the database, the benefit of using higher quality training data outweighs the decrease in the number of training samples. This claim is further supported by experiments (see Section 6). To filter the training data, we choose only those dialogues where the annotated dialogue goal corresponds with the turn-level annotated data. When using the *clean* samples, we omit about 30% of the training data.

To effectively combine all our datasets, we unified the data ontologies. Since the datasets use different naming conventions (e.g., `leaveAt` vs. `leave_at`) and different domain and slot names to describe the same concepts (e.g., `restaurant-food` vs. `restaurant-type`), we

<sup>7</sup>There are also other large-sized task-oriented datasets such as MetalWOZ (Lee et al., 2019), however, their annotation is not detailed enough for our setup.

manually designed a mapping between domain and slot names. Notably, we decided to rename some slots so they use natural language tokens, as we base our model on the GPT-2 LM which is pre-trained on natural language texts (e.g. “`leaveAt`” → “`leave at`”). Our final ontology that unifies all three datasets contains 22 domains and 135 slots.

We use our own implementation of delexicalization, which directly produces our belief state string representation (see Section 3.1 and Example 1).

## 4.3 Training Details

We implement our model in PyTorch (Paszke et al., 2019), based on GPT-2-*small*. It uses 12 layers with a size of 768. For all auxiliary tasks, we use a dropout of 0.1 with label smoothing 0.1. We use the AdamW optimizer (Loshchilov and Hutter, 2019). The finetuning runs for 8 epochs on the MultiWOZ 2.1 data when all the training examples are used, and for the same number of minibatches when using only *clean* samples. The training takes less than one day when using 4 GPUs.

## 4.4 Corpus-based Evaluation

To compare with previous results on MultiWOZ, we evaluate the model performance with a set of corpus-based intrinsic metrics on both versions of the data. For MultiWOZ 2.0, we use the original delexicalization used by compared baselines (Peng et al., 2021; Hosseini-Asl et al., 2020; Zhang et al., 2020b). For MultiWOZ 2.1, we use our own delexicalization. We employ the original evalua-

Method	Average Success	Success w/ DB	Success w/o DB	NLU score	Response appropriateness	Turns
Baseline	69.6	56.8	82.4	4.34	4.18	18.5
Winner	<b>74.8</b>	<b>70.2</b>	79.4	<b>4.54</b>	<b>4.47</b>	18.5
Our submission	72.3	62.0	<b>82.6</b>	4.53	4.41	<b>17.1</b>

Table 3: Human evaluation results obtained during the DSTC9 shared task using Amazon Mechanical Turk. Note that only 4 out of 10 submissions outperformed the Baseline according to the average success metric.

tion scheme by Budzianowski et al. (2018), which provides two metrics – the *inform rate* and the *success rate*. The *inform rate* is the percentage of dialogues in which the system mentioned a name or ID of an entity which does not contradict the current dialogue state and the user’s goal, whereas the *success rate* is the percentage of dialogues in which the system outputted all the requested information. Moreover, we compute BLEU (Papineni et al., 2002) between the generated system utterances and the ground truth to get an approximation of the output fluency.

#### 4.5 ConvLab 2 Evaluation

We use the ConvLab 2 platform (Zhu et al., 2020) for automatic evaluation with a simulated user agent. We run the evaluation component 1,000 times, i.e. on 1,000 simulated conversations. The agent mimics user behavior, interacts with the system under evaluation, and computes multiple metrics: The *complete rate* reflects the ratio of dialogues that are completed, i.e. all the user requests have been met. The *success rate* computes the percentage of dialogues which are successful, meaning the system captures correct informed entities and provides a valid booking if requested. Finally, the *book rate* is the proportion of dialogues where the system was able to book the correct entity (hotel, restaurant, train) if it was asked to. We also compute *precision*, *recall* and *F1 score* for the informed entities and the average number of turns in the dialogue.

#### 4.6 Human Evaluation and Error Analysis

Thanks to our participation in the DSTC9 task-oriented dialogue shared task (Gunasekara et al., 2020; Li et al., 2021), a variant of our model (without pre-training on additional dialogue datasets, see Table 4) was selected for evaluation by human judges on the Amazon Mechanical Turk platform.<sup>8</sup>

<sup>8</sup>The selection was done based on ConvLab 2 performance, but probably used a different version of the tool and thus arrived at different results – the chosen variant is not the best

The judges communicated with the agent in natural language and rated the system afterward with respect to the success/failure of the dialogue, language understanding score, and response appropriateness. Information provided by the system was additionally checked for consistency with the database, and the average of success rates given by the judges and by database grounding is used as the main metric.

In addition to the crowdsourced evaluation, we perform a detailed in-house error analysis based on human interactions with our final system. Expert annotators followed randomly chosen dialogue goals accompanying the MultiWOZ test set and recorded any incorrect system behavior.

## 5 Results

We first discuss quantitative results for both corpus-based and crowdsourced human evaluation, then include a qualitative analysis of the model behavior.

### 5.1 Corpus-based Evaluation on MultiWOZ

Table 1 shows a comparison between our methods and current state-of-the-art systems (cf. Section 2). Since some of the compared methods do not provide results with on MultiWOZ 2.1, we report results on both MultiWOZ 2.0 and MultiWOZ 2.1. As we can see, AuGPT compares favorably to other approaches. The chosen variant of our model is not the best-scoring variant on corpus-based metrics (see Table 4). It was chosen based on the ConvLab evaluation, which may not be optimal for corpus-based evaluation. LABES-S2S produces higher BLEU scores, which would indicate a better fluency of the model, but scores lower on inform and success rates. LAVA, SOLOIST, SimpleTOD, and MD-Sequicity, on the other hand, provide slightly higher inform and success scores while doing worse in terms of fluency.

Table 2 shows a comparison with two other models in the ConvLab evaluation scheme with a simulated user. The compared systems were chosen one according to our own measurements.

[...]  
**user:** *I want to make a booking at sitar tandoori*  
**belief:** restaurant {name = sitar tandoori}  
train {arrive by = 18:00, day = sunday, departure = cambridge, destination = london liverpool street, people = 1}  
**system:** What day would you like the booking for?  
**user:** *the same day, 13:30*  
**belief:** restaurant {day = sunday, name = sitar tandoori, people = 1, time = 13:30}  
train {arrive by = 18:00, day = sunday, departure = cambridge, destination = london liverpool street, people = 1}  
**system:** Booking was successful. The table will be reserved for 15 minutes. Reference number is: 00000200.

Example 2: A dialogue where the system did not ask for necessary information. The user previously booked a train trip for one person (cf. the belief state in the first turn shown). When they went on to book a restaurant, the system took over the number of persons from the train ride and did not ask again.

because they both implement fully trainable end-to-end methods. Our system outperforms both compared systems by a wide margin. Our model is able to perform well not just in a single-turn response generation scenario, but over the course of the whole dialogue. As the example of DAMD shows, this is not always guaranteed.

## 5.2 Human Evaluation

We show the DSTC9 human evaluation results in Table 3. Our model outperformed the baseline on the main average success metric (along with only 3 other submissions out of 10). We did not match the winning submission on average success, but we achieved the best overall success rate based on human judgments (without checking consistency with the database). This suggests that the main drawback of our model lies in database grounding. This could have been caused by our new delexicalization (Section 4.2). Finally, our model needed the smallest number of turns on average to complete the dialogues.

Note that the model variant chosen for human evaluation did not include pre-training on additional data (cf. Section 4.6 and Footnote 8), which could improve performance in human evaluation.

## 5.3 Error Analysis

Our expert annotators evaluated 130 dialogues in total, 50 of which contained at least one error. However, in most cases, the system was able to recover from the errors, resulting in an overall success rate

**user :** I am looking for a cheap *chinse* restaurant.  
**belief:** restaurant {food = chinese, price range = cheap}  
**system:** I have 4 cheap chinese restaurants. What area of town would you like to dine in?  
**user:** downtown  
**belief:** restaurant {area = centre, food = chinese, price range = cheap}  
**system:** I have 3 cheap chinese restaurants in the centre. Would you like to book at Charlie Chan?  
...

Example 3: A snippet from a challenging but successful dialogue. The system deals with input noise (“chinse”) as well as with paraphrases (“downtown”).

of 86.9% (i.e., 17 unsuccessful dialogues). The purpose of this analysis was to identify different types of errors occurring during full dialogues.

By far the most common error (21 counts) were *hallucinated values*, i.e., lack of grounding for some of the information provided (see the end of Example 2). Another frequent error type is *missing information* (5 counts), i.e., not asking for information that is required (and e.g. reusing information from a different domain without user confirmation). Example 2 also demonstrates another common error type, which is *bad domain* (4 counts). Here, the system does not react to the user’s request for a different search (hotels instead of attractions in the example). This might be caused by a less frequent wording of the request in the given context, and usually gets resolved by rephrasing the request.

The analysis also found many examples of correct system behavior in non-trivial cases. As illustrated in Example 3, the model is able to deal with paraphrases and is robust to a certain amount of noise in the data. Specifically, it handles typos, various time formats, etc. Interaction between domains is also successful in most cases – the system is able to resolve references to another domain’s belief state (e.g., make a booking for the same group of people as done before in a different venue).

## 6 Ablation Study

We tested many variants of our method with different combinations of components to evaluate their contributions. The results are presented in Table 4. Namely, we are interested in the following components: (1) unlikelihood loss, (2) auxiliary tasks, (3) data augmentation, (4) modified consistency task and (5) unclean data filtering.

We can see that all proposed contributions which are a part of our final system, except for the unlikelihood training, have a positive effect on the

method	MultiWOZ 2.1			ConvLab 2						
	inf	suc	BLEU	comp	suc	book	P	R	F1	turns
<b>AuGPT</b>	83.5	67.3	17.2	<b>89.4</b>	<b>60.1</b>	85.7	64.5	<b>82.1</b>	70.3	14.6
w/o. unlikelihood	84.1	66.9	17.1	89.2	59.3	<b>90.8</b>	63.9	81.6	69.5	14.6
w/o. clean	81.9	64.0	15.8	85.0	57.7	85.6	65.6	79.1	69.6	14.5
w/o. unlikelihood, w/o. clean	<b>86.5</b>	<b>69.1</b>	<b>17.5</b>	85.9	58.4	81.3	62.2	79.8	67.5	<b>14.1</b>
w. all auxiliary	83.1	66.2	17.0	88.7	59.2	86.0	64.6	81.1	69.9	14.4
w/o. pre-training	81.0	62.7	15.1	88.1	59.8	83.7	<b>68.1</b>	80.9	72.1	15.6
w/o. back-translations	79.8	61.7	15.2	88.9	58.2	87.4	68.0	81.6	<b>72.2</b>	14.9
w. old consistency	81.4	65.8	17.0	85.5	57.8	86.0	65.2	80.0	69.8	14.6
w/o. consistency	81.9	64.5	16.3	86.4	57.1	84.1	66.3	81.2	70.9	14.6

Table 4: Ablation study (inf = inform, suc = success, book = book rate; see Section 4.4). The model version with the best ConvLab 2 success rate is chosen as our best model. Variants are denoted with their respective modifications compared to the default: “w/o. unlikelihood” = unlikelihood loss was not used; “w/o. clean” uses all training samples as opposed to using only the ones consistent with the database; “w/o. pre-training” = Taskmaster-1 and Schema-Guided datasets were not used for training (this variant was selected for human evaluation); “all auxiliary” = using two additional auxiliary tasks (see the Method section); “w/o. consistency” = dialogue consistency task is not used; “old consistency” refers to the consistency task by Peng et al. (2021) (see the Section 3.2).

system performance. In the ConvLab evaluation, our final system performs best. Removing either pre-training or back-translations decreases BLEU, inform and success rates substantially. Furthermore, we notice the positive effect of using our improved consistency detection task over the one used in SOLOIST (Peng et al., 2021), which in turn scores better than no consistency detection.

Training on all data as opposed to using only “clean” samples clearly reduces performance. On the other hand, unlikelihood training improves performance only in ConvLab while causing a performance drop in corpus-based metrics. This can be caused by the fact that the unlikelihood training promotes diversity and reduces repetitions on the token level, and thus does not play well with corpus-based evaluation. We did not notice any increase in performance when the user intent prediction and system action prediction auxiliary tasks were used (cf. Section 3.2). The reason for this behavior could be that the model learns to represent the actions well enough implicitly, without the need for these additional objectives. Therefore, these tasks are not a part of our final model.

## 7 Conclusions & Future Work

We present a dialogue modeling pipeline based on the pre-trained GPT-2 language model. AuGPT uses modified training objectives and employs data augmentation to increase the diversity of generated utterances. Our experiments show that the proposed approach outperforms baselines and is

competitive with state of the art on the MultiWOZ dataset. We also run a series of ablation experiments to assess the individual contributions of the modifications. According to our detailed ablation study, training data augmentation using back-translation via multiple languages and a modified auxiliary training objective for dialogue consistency detection are the features that contribute most to our system’s performance. Additionally, we perform a qualitative analysis of the outputs to give a better insight into our model behavior.

In the future, we plan to construct a latent representation of the belief state and optimize it jointly with the language model. We will replace the deterministic lexicalization with a trainable alternative, and possibly even integrate the database module into the model. To improve the transfer to new domains, we will learn a domain embedding and optimize it jointly with the model, unifying all datasets.

## Acknowledgments

This work was supported by the Charles University GAUK grant No. 302120, the SVV project No. 260575, and the Charles University project PRIMUS/19/SCI/10. Jonáš Kulhánek was supported by the European Regional Development Fund under the project Robotics for Industry 4.0 (reg. no. CZ.02.1.01/0.0/0.0/15\_003/0000470). Additional computational resources were supplied by the project “e-Infrastruktura CZ” (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures.



## References

- Paweł Budzianowski and Ivan Vulić. 2019. [Hello, it's GPT-2 - how can I help you? Towards the use of pre-trained language models for task-oriented dialogue systems](#). In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. [MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium.
- Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. [Taskmaster-1: Toward a realistic and diverse dialog dataset](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4516–4525, Hong Kong, China.
- 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.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding back-translation at scale](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium.
- Arash Einolghozati, Sonal Gupta, Mrinal Mohit, and Rushin Shah. 2019. [Improving robustness of task oriented dialog systems](#). In *3rd Conversational AI Workshop at 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.
- Henry Elder, Robert Burke, Alexander O'Connor, and Jennifer Foster. 2020. [Shape of synth to come: Why we should use synthetic data for English surface realization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7465–7471, Online.
- 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, Marseille, France.
- Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. [Key-value retrieval networks for task-oriented dialogue](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany.
- Christian Federmann, Oussama Elachqar, and Chris Quirk. 2019. [Multilingual whispers: Generating paraphrases with translation](#). In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 17–26, Hong Kong, China.
- Jianfeng Gao, Michel Galley, and Lihong Li. 2019. [Neural approaches to conversational AI: Question answering, task-oriented dialogues and social chatbots](#). volume 13 of *Foundations and Trends in Information Retrieval*.
- Claudio Greco, Barbara Plank, Raquel Fernández, and Raffaella Bernardi. 2019. [Psycholinguistics meets continual learning: Measuring catastrophic forgetting in visual question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3601–3605, Florence, Italy.
- Chulaka Gunasekara, Seokhwan Kim, Luis Fernando D'Haro, Abhinav Rastogi, Yun-Nung Chen, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, Dilek Hakkani-Tür, Jinchao Li, Qi Zhu, Lingxiao Luo, Lars Liden, Kaili Huang, Shahin Shayandeh, Runze Liang, Baolin Peng, Zheng Zhang, Swadheen Shukla, Minlie Huang, Jianfeng Gao, Shikib Mehri, Yulan Feng, Carla Gordon, Seyed Hossein Alavi, David Traum, Maxine Eskenazi, Ahmad Beirami, Eunjoon, Cho, Paul A. Crook, Ankita De, Alborz Geramifard, Satwik Kottur, Seungwhan Moon, Shivani Poddar, and Rajen Subba. 2020. [Overview of the ninth dialog system technology challenge: DSTC9](#). *arXiv preprint arXiv:2011.06486*.
- Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. [End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text degeneration](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.
- Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. [A simple language model for task-oriented dialogue](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Lifeng Jin, David King, Amad Hussein, Michael White, and Douglas Danforth. 2018. [Using paraphrasing](#)

- and memory-augmented models to combat data sparsity in question interpretation with a virtual patient dialogue system. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 13–23, New Orleans, Louisiana.
- Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. **Neural AMR: Sequence-to-sequence models for parsing and generation**. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada.
- Sungjin Lee, Hannes Schulz, Adam Atkinson, Jianfeng Gao, Kaheer Suleman, Layla El Asri, Mahmoud Adada, Minlie Huang, Shikhar Sharma, Wendy Tay, and Xiujun Li. 2019. **Multi-domain task-completion dialog challenge**.
- Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. **Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Melbourne, Australia.
- Jinchao Li, Qi Zhu, Lingxiao Luo, Lars Liden, Kaili Huang, Shahin Shayandeh, Runze Liang, Baolin Peng, Zheng Zhang, Swadheen Shukla, Ryuichi Takanobu, Minlie Huang, and Jianfeng Gao. 2021. **Multi-domain task completion dialog challenge II at DSTC9**. In *DSTC9 Workshop at AAI*, Online.
- Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. **Don’t say that! Making inconsistent dialogue unlikely with unlikelihood training**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728, Online.
- Chin-Yew Lin. 2004. **Rouge: A package for automatic evaluation of summaries**. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain.
- Ilya Loshchilov and Frank Hutter. 2019. **Decoupled weight decay regularization**. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.
- Nurul Lubis, Christian Geisshauser, Michael Heck, Hsien-chin Lin, Marco Moresi, Carel van Niekerk, and Milica Gasic. 2020. **LAVA: Latent action spaces via variational auto-encoding for dialogue policy optimization**. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 465–479, Barcelona, Spain (Online).
- Dominik Macháček, Jonáš Kratochvíl, Sangeet Sagar, Matúš Žilinec, Ondřej Bojar, Thai-Son Nguyen, Felix Schneider, Philip Williams, and Yuekun Yao. 2020. **ELITR non-native speech translation at IWSLT 2020**. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 200–208, Online.
- Nitin Madnani and Bonnie J. Dorr. 2010. **Generating phrasal and sentential paraphrases: A survey of data-driven methods**. *Computational Linguistics*, 36(3):341–387.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. **BLEU: a method for automatic evaluation of machine translation**. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. **PyTorch: An imperative style, high-performance deep learning library**. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.
- Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. **Soloist: Building task bots at scale with transfer learning and machine teaching**. *Transactions of the Association for Computational Linguistics*, 9:807–824.
- Libo Qin, Xiao Xu, Wanxiang Che, Yue Zhang, and Ting Liu. 2020. **Dynamic fusion network for multi-domain end-to-end task-oriented dialog**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6344–6354, Online.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. **Language models are unsupervised multitask learners**. Technical report, OpenAI.
- 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*, pages 8689–8696, New York, NY, USA.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. **Improving neural machine translation models with monolingual data**. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany.
- 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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Sean Welleck, Ilya Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. [Neural text generation with unlikelihood training](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.
- Tsung-Hsien Wen, Milica Gašić, Dongho Kim, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. [Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking](#). In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 275–284, Prague, Czech Republic.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. [A network-based end-to-end trainable task-oriented dialogue system](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain.
- Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. [ToD-BERT: Pre-trained natural language understanding for task-oriented dialogues](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 917–929, Online.
- S. Young, M. Gašić, B. Thomson, and J. D. Williams. 2013. [POMDP-based statistical spoken dialog systems: A review](#). *Proceedings of the IEEE*, 101(5):1160–1179.
- Yichi Zhang, Zhijian Ou, Min Hu, and Junlan Feng. 2020a. [A probabilistic end-to-end task-oriented dialog model with latent belief states towards semi-supervised learning](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9207–9219, Online.
- Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020b. [Task-oriented dialog systems that consider multiple appropriate responses under the same context](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9604–9611, New York, NY, USA.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020c. [DIALOGPT : Large-scale generative pre-training for conversational response generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online.
- Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. [ConvLab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 142–149, Online.
- Vilém Zouhar, Michal Novák, Matúš Žilinc, Ondřej Bojar, Mateo Obregón, Robin L. Hill, Frédéric Blain, Marina Fomicheva, Lucia Specia, and Lisa Yankovskaya. 2021. [Backtranslation feedback improves user confidence in MT, not quality](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 151–161, Online.

## A Additional Results

### A.1 Detailed Error Analysis

Our expert annotators evaluated 130 dialogues in total, 50 of which contained at least one error. However, in most cases, the system was able to recover from the errors, resulting in an overall success rate of 86.9% (i.e., 17 unsuccessful dialogues). The purpose of this analysis was to identify different types of errors occurring during full dialogues. The annotators were familiar with the model architecture and were instructed to categorize the errors according to the cause of the problem. Specifically, they identified which component caused the respective error and annotators categorized the errors into more specific types.

The overall results are given in Table 5. We observe that the most common reason for a failed dialogue is an error related to the belief state (30 errors, 10 failed dialogues). Also, although policy errors happen relatively often (21x), they rarely cause the whole dialogue to fail (2 dialogues). We observe that we have a slightly higher number of successful dialogues compared to the 82.6% success rate (without checking database consistency) found in human evaluation (cf. Table 3). The most likely cause is that our expert annotators were more motivated to recover from erroneous system behavior and finish the dialogue.

Fine-grained error types identified by annotators are given in Table 6 and Examples 2, 4 and 3. To extend the analysis from Section 5.3, we include another frequent error type – *missing information* (5 counts), i.e., not asking for information that is required (Example 4). In this case, the system uses information from a different domain without the user explicitly confirming this. A most probable cause of this is that most instances of the training data carry over the information.

### A.2 Individual Component Analysis

We have conducted additional tests to obtain a deeper insight into each component’s performance – DST and NLG. We have evaluated the accuracy of the generated belief states. Joint accuracy, slot accuracy, and F1 score were used. Joint accuracy gives the percentage of successfully generated belief states – with no error. Slot accuracy, on the other hand, is the average accuracy of correctly predicting the value for a domain-slot pair. To evaluate NLG, we compared the end-to-end system where the generated belief state is used to query

<p><b>user:</b> <i>I'm looking for a theatre in Cambridge</i> belief: attraction {type = theatre} <b>system:</b> There are 5 theatres in Cambridge. Is there an area of town you prefer? <b>user:</b> <i>no I don't care</i> belief: attraction {type = theatre} <b>system:</b> I recommend Adc Theatre in the centre. Would you like more information? <b>user:</b> <i>no, I'm also looking for some place to stay</i> belief: attraction {type = theatre} <b>system:</b> I have 5 theatres in the centre. Do you have a price range? <b>user:</b> <i>I'm looking for a place to stay</i> belief: attraction {type = theatre} <b>system:</b> There are 5 theatres in the centre. Do you have a preference for area? <b>user:</b> <i>I'm looking for a hotel</i> belief: hotel {type = hotel} attraction {type = theatre} <b>system:</b> <i>There are 9 hotels in the north. Do you have a price range?</i></p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Example 4: Dialogue sample with a bad domain focus and a hallucination. First, the system ignores that the user switched from searching for a theater to searching for a hotel. After accepting the new domain, the system replies with hotels “in the north” even though the user did not specify.

the database and generate the response with a variant of the pipeline, where the ground-truth belief state and/or ground-truth database result counts were used. The BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004) scores were used for evaluation.

In Table 7, we can see the performance of each individual component of the system. One can notice that the performance of NLG is not decreased when we use the generated belief state instead of the oracle belief state. Since the belief state prediction is not perfect, this suggests that the model does not actually need belief states for generating the delexicalized response. However, when the real database result counts are used instead of oracle database result counts, the performance decreases, which implies that the database result counts are important for NLG.

	Erroneous Dialogues	BS Error	DB Error	Policy Error	Other Error
All dialogues	50	30	4	21	6
Unsuccessful dialogues	17	10	3	2	2

Table 5: Interactive analysis performed by human evaluators using 130 prepared dialogue goals. 17 of these dialogues contained an error that caused the dialogue to fail. We show summary statistics regarding the number of respective error sources (BS = belief state, DB = database). Note that some of the dialogues contain more than one error.

Type	Count	Source	Description
Hallucinated values	21	BS/Policy	Used a slot value in the reply that is not grounded in the DB nor in the context
Wrong lexicalization	6	Policy	Repeats the same value in a list of choices during lexicalization
Missing information	5	Policy	Makes booking while not all information is specified
Ignored input	5	BS	Keeps asking for information that was provided
Bad domain	4	BS	Fails to focus on the correct domain
False response	4	Policy	States a different value of a slot than the value stored in DB
Repeated output	3	Policy	Repeats the same slot twice on the output
Failed booking	3	DB/Policy	Booking was unsuccessful due to DB mismatch
Other	10	BS/DB/P/Oth	(Various rare errors that could not be categorized)

Table 6: Distribution of the most common error types encountered during the human evaluation of 130 dialogues. Absolute counts of errors in the 50 erroneous dialogues are shown. The total error count is 61 as some dialogues contained multiple errors. The most likely source of the error (cf. Table 5) and a short description are given for each type.

fine-tuned on	oracle		DST			NLG	
	bs	db	joint acc.	slot acc.	F1	BLEU	ROUGE-L
MW 2.0	✗	✗	54.1	97.2	90.0	17.2	39.0
	✗	✓				17.4	39.3
	✓	✓				17.4	39.2
MW 2.1	✗	✗	56.5	97.2	90.6	17.4	38.6
	✗	✓				17.6	38.8
	✓	✓				17.6	38.8

Table 7: Performance of DST and NLG components. Joint and slot accuracies, as well as slot values F1 score, are used to evaluate DST. For NLG, BLEU and ROUGE-L metrics are used. Apart from using the generated belief states and database counts, we also evaluate the components with oracle values. Note that models were pre-trained on Taskmaster-1 and Schema-Guided Dialogue datasets.