Pretraining the Noisy Channel Model for Task-Oriented Dialogue

Qi Liu[2*], **Lei Yu**[1], **Laura Rimell**[1], **and Phil Blunsom**[1,2]
[1]DeepMind, United Kingdom [2]University of Oxford, United Kingdom
qi.liu@cs.ox.ac.uk
{leiyu,laurarimell,pblunsom}@google.com

Abstract

Direct decoding for task-oriented dialogue is known to suffer from the explaining-away effect, manifested in models that prefer short and generic responses. Here we argue for the use of Bayes' theorem to factorize the dialogue task into two models, the distribution of the context given the response, and the prior for the response itself. This approach, an instantiation of the noisy channel model, both mitigates the explaining-away effect and allows the principled incorporation of large pretrained models for the response prior. We present extensive experiments showing that a noisy channel model decodes better responses compared to direct decoding and that a two-stage pretraining strategy, employing both open-domain and task-oriented dialogue data, improves over randomly initialized models.

1 Introduction

Task-oriented dialogue agents provide a conversational interface to assist users in accomplishing specific goals, such as finding a restaurant or booking a hotel (Seneff and Polifroni, 2000; Raux et al., 2005; Budzianowski et al., 2018; Peng et al., 2020a). Increasing demand from industry for natural language assistants and scalable customer service solutions has recently been driving a renaissance in the development of task-oriented dialogue models. In addition, the specification of explicit dialogue agent goals, afforded by the task-oriented paradigm, makes such research easier to ground and evaluate than open-domain chatbots.

Current research on task-oriented dialogue is dominated by monolithic sequence-to-sequence models that directly parameterize the conditional distribution of the response given the prior dialogue context. However, this monolithic approach conflates the task-specific and language-general aspects of dialogue, and adversely favors short and generic responses (Bao et al., 2020) due to the explaining-away effect (Klein and Manning, 2002).

Here we pursue an alternative to the direct model. Using Bayes' rule allows us to factorize the probability of the response given the context $p(\mathcal{R}|\,\mathcal{C})$ into a language model $p(\mathcal{R})$ and a context model $p(\mathcal{C}|\,\mathcal{R})$.[1] Within natural language processing (NLP), this approach is traditionally known as the noisy channel model (Shannon, 1948), and has recently seen renewed interest with its successful application to neural machine translation (Yu et al., 2017, 2020; Yee et al., 2019).

We hypothesize that the noisy channel reformulation is advantageous for dialogue because the factorization enables each sub-module to specialize in a dialogue sub-task. In particular, the context conditional model can help to discount short and generic responses and mitigate the explaining-away effect, while the language model helps ensure that responses are natural. We find that a noisy channel model with the same number of parameters as a direct model achieves better accuracy on three task-oriented dialogue datasets. Moreover, a larger noisy channel model can be trained with the same hardware, by training the sub-modules separately, yielding additional improvements.

It has become common in recent years to pretrain dialogue models on large text data, either general text (Peng et al., 2020b; Budzianowski and Vulić, 2019; Wu et al., 2020a) or dialogue-structured data (Roller et al., 2020; Adiwardana et al., 2020), such as tweets and Reddit posts. We utilise a similar strategy with Reddit data and find

[*]Work completed during an internship at DeepMind.

[1]Here we abstract away from the prediction of belief states and dialogue acts, which also form part of our generative model; see Section 3 for details.

Transactions of the Association for Computational Linguistics, vol. 9, pp. 657–674, 2021. https://doi.org/10.1162/tacl_a_00390
Action Editor: Wenjie (Maggie) Li. Submission batch: 2/2021; Revision batch: 2/2021; Published 7/2021.

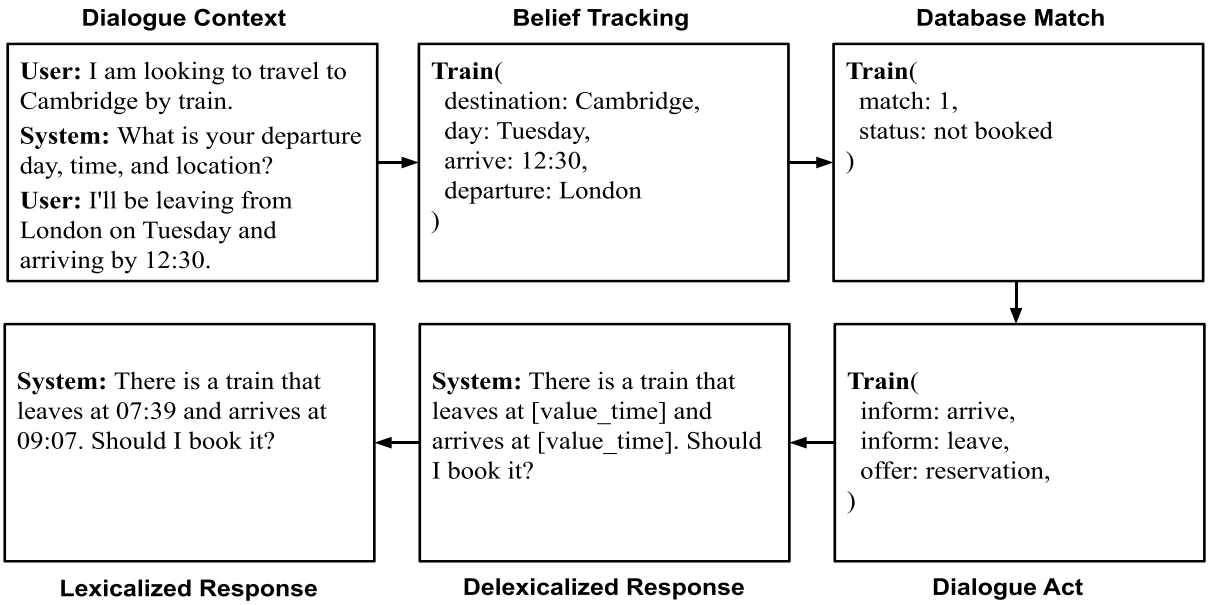


Figure 1: The data flow of one turn in a task-oriented dialogue for train booking from MultiWOZ.

that the benefits of pretraining to the noisy channel model are similar to those for the direct model. Further, we evaluate transfer across task-oriented dialogue datasets by implementing a second pretraining stage using Taskmaster (Byrne et al., 2019) and Schema-Guided Dialogue (Rastogi et al., 2020) as training data, before fine-tuning on our final tasks.

We evaluate the algorithm on three datasets, MultiWOZ 2.0 (Budzianowski et al., 2018), CamRest676 (Wen et al., 2017a), and SMCalFlow (Andreas et al., 2020), demonstrating that the noisy channel approach is robust to different dialogue schema annotations used across datasets. Further analysis demonstrates that the noisy channel models can decode responses with similar lengths and Zipf scores compared to ground-truth responses and reduce the likelihood of falling into repetition loops (Holtzman et al., 2019).

2 A Seq-to-Seq Dialogue Model

In this section, we introduce a discriminative sequence-to-sequence model for task-oriented dialogue. The traditional sequence of steps needed to produce a system turn in a task-directed dialogue is shown in Figure 1, with an example from MultiWOZ 2.0 (Budzianowski et al., 2018). Given a dialogue context containing previous user and system utterances, the dialogue system first predicts a belief state, consisting of a set of slot-value pairs (e.g., `destination: Cambridge`), to capture user intent. To ground the system with external information, the belief state can be converted into a database query in order to retrieve relevant information, such as the number of matches and booking information. Next, the system predicts a set of dialogue acts, representing the abstract meaning of the proposed dialogue response (Austin, 1975). Finally, a delexicalized dialogue response is generated, where slot values are replaced by generic placeholders, such as `value_time` for a train departure time, in order to reduce lexical variation. The delexicalized response can be converted to a lexicalized response in post-processing by filling in the slot values based on belief states and database information.

We use the MultiWOZ schema for illustration in Sections 2 and 3, but our models easily generalize to different schema annotations (e.g., datasets without annotated dialogue acts [Andreas et al., 2020]).

Because it is well known that pipelined models tend to suffer from error propagation, many NLP tasks have been reformulated in recent years as end-to-end text-to-text transformations (Raffel et al., 2020; Brown et al., 2020). State-of-the-art task-oriented dialogue systems have followed this approach (Hosseini-Asl et al., 2020; Peng et al., 2020b). We represent the example from Figure 1 as follows, serializing turns and using special start and end tokens to encapsulate each data field:

Context: [c] I am looking to . . . [/u] What is your . . . [/r] I'll be leaving . . . [/u] [/c]
Belief: [b] [train] destination Cambridge, day Tuesday, arrive 12:30, departure London [/b]
Database: [db] [train] match 1, status not booked [/db]
Act: [a] [train] inform arrive, inform leave, offer reservation [/a]
Response: [r] There is a train that leaves at [value_time] and arrives at [value_time]. Should I book it? [/r]

Given this text representation, the direct discriminative approach models $p(\mathcal{B}, \mathcal{A}, \mathcal{R} | \mathcal{C})$, where $\mathcal{C}$, $\mathcal{B}$, $\mathcal{A}$, and $\mathcal{R}$ represent dialogue context, belief state, dialogue act, and delexicalized response, respectively.[2] We use the serialized text of the dialogue context as input, and the concatenation of belief state, dialogue act, and response as target output, making the task amenable to the application of an autoregressive sequence-to-sequence model. $\mathcal{B}$, $\mathcal{A}$, and $\mathcal{R}$ can be generated sequentially with direct decoding methods, such as greedy decoding and beam search. We use a sequence-to-sequence Transformer (Vaswani et al., 2017) to implement $p(\mathcal{B}, \mathcal{A}, \mathcal{R} | \mathcal{C})$. This distribution will also be used to build the noisy channel model in Section 3.

3 Noisy Channel Model for Dialogue

While direct decoding is an effective approach for decoding belief states (Hosseini-Asl et al., 2020), it may be sub-optimal for generating responses. First, it favors short and generic responses (Bao et al., 2020). As a result, the decoded responses are bland and lack diversity (Li et al., 2016). Second, it suffers from the explaining-away effect (Klein and Manning, 2002), where inputs are ''explained-away'' by highly predictive output prefixes. For example, if there is one hotel matching the user's intent as encoded in the belief state, the model is nevertheless prone to decoding ''no'' given the output prefix ''there is'', ignoring the input information.

In this work, we propose using the neural noisy channel model (Yu et al., 2017) to mitigate the above problems for response generation. Given an input sequence x and output sequence y, the noisy channel formulation (Shannon, 1948) uses Bayes' rule to rewrite the model $p(y|x)$ as $\frac{p(x|y)p(y)}{p(x)} \propto p(x|y)p(y)$. It was originally applied to speech recognition, where $p(y|x)$ is a conditional model of the source text given a noisy observation. The *channel model* $p(x|y)$ estimates the probability of the observation given the source, while $p(y)$ is an unconditional *language model* (or *source model*), which can be trained on unpaired data. More recently it has been applied to machine translation, where y is a translation of input text x.

Abstracting away from belief states and dialogue acts, for task-oriented dialogue we want to estimate $p(\mathcal{R} | \mathcal{C})$, the probability of a response given a context. The channel model $p(\mathcal{C} | \mathcal{R})$, given a response, predicts a distribution over contexts which might have elicited that response. The source model $p(\mathcal{R})$ is an unconditional language model. In this extension of the noisy channel approach to task-oriented dialogue, the ''channel'' can be understood as connecting dialogue contexts with suitable responses.

For the full task, we develop a noisy channel model for $p(\mathcal{B}, \mathcal{A}, \mathcal{R} | \mathcal{C})$. Using the chain rule, $p(\mathcal{B}, \mathcal{A}, \mathcal{R} | \mathcal{C}) = p(\mathcal{B} | \mathcal{C}) \cdot p(\mathcal{A}, \mathcal{R} | \mathcal{C}, \mathcal{B})$. Following Hosseini-Asl et al. (2020), we use the direct model described in Section 2 to parameterize $p(\mathcal{B} | \mathcal{C})$ and decode $\mathcal{B}$, which our preliminary experiments confirmed to be advantageous.

We use the noisy channel formulation to parameterize $p(\mathcal{A}, \mathcal{R} | \mathcal{C}, \mathcal{B})$. Using Bayes' rule, $p(\mathcal{A}, \mathcal{R} | \mathcal{C}, \mathcal{B}) \propto p(\mathcal{C}, \mathcal{B} | \mathcal{A}, \mathcal{R}) \cdot p(\mathcal{A}, \mathcal{R})$. The channel model $p(\mathcal{C}, \mathcal{B} | \mathcal{A}, \mathcal{R})$ and source model $p(\mathcal{A}, \mathcal{R})$ are implemented as Transformers.

We choose to use the noisy channel formulation for decoding $\mathcal{A}$ based on preliminary experiments that showed improved overall accuracy over direct decoding, possibly because poor dialogue act prediction by the direct model led to worse quality responses. The serialized text of $\mathcal{A}$ and $\mathcal{R}$ are concatenated during training, and the decoded sequence is split into $\mathcal{A}$ and $\mathcal{R}$ with the special start/end tokens during decoding.

We suggest that the noisy channel model has three advantages over the direct model for response generation: (1) The channel model can penalize short and generic responses. Such responses can be mapped to a large number of contexts, resulting in a flat distribution over contexts. This leads to a lower channel model score for short and generic responses (Zhang et al., 2020b). (2) The channel model ensures that $(\mathcal{A}, \mathcal{R})$ must explain the corresponding $(\mathcal{C}, \mathcal{B})$, alleviating the explaining-away effect (Yu et al., 2017). (3) The

[2] We do not model the probabilities of database state or lexicalized response, as these are deterministic given the belief state and delexicalized response, respectively.

source model, an unconditional distribution over $\mathcal{A}$ and $\mathcal{R}$, can make use of abundant non-dialogue textual data for pretraining, further improving the fluency of generated sequences (Brants et al., 2007). We leave exploration of this last advantage for future work, as we pretrain all sub-modules with the same data.

3.1 Decoding

Because exact decoding from the noisy channel model $\arg\max_{\mathcal{A},\mathcal{R}} p(\mathcal{C},\mathcal{B}|\,\mathcal{A},\mathcal{R}) \cdot p(\mathcal{A},\mathcal{R})$[3] is computationally intractable, we experiment with two approximation methods, noisy channel reranking and noisy channel online decoding. Since these methods rely on $p(\mathcal{A},\mathcal{R}|\,\mathcal{C},\mathcal{B})$ as a proposal distribution for approximation, and both $p(\mathcal{A},\mathcal{R}|\,\mathcal{C},\mathcal{B})$ and $p(\mathcal{B}|\,\mathcal{C})$ are parameterized with the direct model introduced in Section 2, our noisy channel model therefore has three sub-modules: a direct model $p(\mathcal{B},\mathcal{A},\mathcal{R}|\,\mathcal{C})$, a channel model $p(\mathcal{C},\mathcal{B}|\,\mathcal{A},\mathcal{R})$, and a source model $p(\mathcal{A},\mathcal{R})$.

Noisy Channel Reranking: Noisy channel reranking first decodes $\mathcal{B}$ and then continues decoding a list $\mathcal{S}$ of $(\mathcal{A},\mathcal{R})$ pairs by beam search with the direct model, prior to utilizing the noisy channel model to rerank $(\mathcal{A},\mathcal{R})$ pairs. In particular, during beam search, partial sequences are expanded and pruned with $p(\mathcal{A},\mathcal{R}|\,\mathcal{C},\mathcal{B})$ (from the direct model in Section 2). The pairs after decoding are reranked using the following model combination:

$$\begin{aligned}(\mathcal{A}',\mathcal{R}') = \arg\max_{(\mathcal{A},\mathcal{R})\in\mathcal{S}} &\log p(\mathcal{A},\mathcal{R}|\,\mathcal{C},\mathcal{B}) + \\ &\lambda_1 \cdot \log p(\mathcal{C},\mathcal{B}|\,\mathcal{A},\mathcal{R}) + \\ &\lambda_2 \cdot \log p(\mathcal{A},\mathcal{R}) + \\ &\lambda_3 \cdot |\,\mathcal{A},\mathcal{R}|,\end{aligned} \quad (1)$$

where $|\,\mathcal{A},\mathcal{R}|$ denotes the length of $(\mathcal{A},\mathcal{R})$, and λ_1, λ_2 and λ_3 are hyperparameters. Besides the channel model $p(\mathcal{C},\mathcal{B}|\,\mathcal{A},\mathcal{R})$ and the source model $p(\mathcal{A},\mathcal{R})$, we additionally use the direct model $p(\mathcal{A},\mathcal{R}|\,\mathcal{C},\mathcal{B})$ and a length bias $|\,\mathcal{A},\mathcal{R}|$ to encourage responses with high direct model likelihood and discourage short responses, respectively.

Noisy Channel Online Decoding: In contrast to reranking, online decoding applies the noisy channel model during beam search for pruning partial sequences, thus exploring a larger search space.

Algorithm 1: Online decoding for the noisy channel.

Input : Context $\mathcal{C}$
Output: Belief, act and response $(\mathcal{B},\mathcal{A},\mathcal{R})$
Decode $\mathcal{B}$ given $\mathcal{C}$ with $p(\mathcal{B}|\,\mathcal{C})$
Beam: $\mathcal{S} = \{([a])\}$
while end($\mathcal{S}$) is **False do**
  $\mathcal{S}' = \varnothing$
  for $\mathcal{O}$ **in** $\mathcal{S}$ **do**
    if $\mathcal{O}$.last() is [/r] **or** $|\,\mathcal{O}| > l$ **then**
      $\mathcal{S}'$.add($\mathcal{O}$)
      continue
    end
    Get k_1 tokens $o^1,\ldots,o^{k_1}$ from the direct model $p(O_{|\,\mathcal{O}|+1}|\,\mathcal{C},\mathcal{B},\mathcal{O})$
    for o^i **in** $(o^1,\ldots,o^{k_1})$ **do**
      $\mathcal{S}'$.add($(\mathcal{O},o^i)$)
    end
  end
  $\mathcal{S} = \underset{\mathcal{O}\in\mathcal{S}'}{\text{top\_k}_2} \log p(\mathcal{O}|\,\mathcal{C},\mathcal{B}) + \lambda_1 \cdot \log p(\mathcal{C},\mathcal{B}|\,\mathcal{O}) + \lambda_2 \cdot \log p(\mathcal{O}) + \lambda_3 \cdot |\,\mathcal{O}|$
end
Select $\mathcal{O}\in\mathcal{S}$ with the largest score using Eq. 1 and return $(\mathcal{B},\mathcal{A},\mathcal{R})$

As shown in Algorithm 1, we first decode the belief state with $p(\mathcal{B}|\,\mathcal{C})$, which comes from the direct model in Section 2. Then, starting with a beam $\mathcal{S}$ containing a single sequence [a] (the dialogue act start token), we continuously expand the sequences in $\mathcal{S}$ until end($\mathcal{S}$) is met, namely, all sequences in $\mathcal{S}$ either end with [/r] or have lengths larger than l. In each iteration, we first expand the sequences in the beam, then prune the expanded beam. To expand a partial act and response sequence (denoted as $\mathcal{O}$ in Algorithm 1), a naive way is to use the noisy channel model to score $|V|$ (the vocabulary size) possible expansions, which is computationally expensive. Instead, we use the probability of the next token $p(O_{|\,\mathcal{O}|+1}|\,\mathcal{C},\mathcal{B},\mathcal{O})$ (where $|\,\mathcal{O}|$ denotes the length of $\mathcal{O}$) to select k_1 candidates to be scored by the noisy channel model. This next token probability is from the direct model introduced in Section 2. One straightforward way to select k_1 expansions from $p(O_{|\,\mathcal{O}|+1}|\,\mathcal{C},\mathcal{B},\mathcal{O})$ is using the top-k maximization, but we can also take advantage of the advances in sampling from a categorical distribution for text generation (e.g., top-k sampling Fan et al., 2018 and nucleus sampling [Holtzman

[3] Although exact decoding is also computationally intractable for the direct model, approximating $\arg\max_{\mathcal{B}} p(\mathcal{B}|\,\mathcal{C})$ is well-studied, e.g., beam search. The decoding for $\mathcal{B}$ is therefore omitted here.

et al., 2019]). After the expansion, we prune the expanded beam $\mathcal{S}'$ to obtain a smaller beam with k_2 partial sequences based on the model combination in Eq. 1. Compared to noisy channel reranking, online decoding applies the noisy channel model during beam search, which is potentially less biased towards the direct model.

In summary, we note that beam search for both the direct model and the online decoding for our noisy channel model decodes $(\mathcal{B}, \mathcal{A}, \mathcal{R})$ autoregressively. Thus both approaches are end-to-end models for task-oriented dialogue. The key difference is that noisy channel online decoding uses Eq. 1 for pruning, while the direct model uses $p(\mathcal{A}, \mathcal{R} | \mathcal{C}, \mathcal{B})$.

4 Model and Pretraining

We use three Transformer (Vaswani et al., 2017) networks to parameterize the direct model $p(\mathcal{B}, \mathcal{A}, \mathcal{R} | \mathcal{C})$, the channel model $p(\mathcal{C}, \mathcal{B} | \mathcal{A}, \mathcal{R})$ and the source model $p(\mathcal{A}, \mathcal{R})$, respectively. The input to each Transformer is the sum of four embeddings: word embeddings, position embeddings, role embeddings (user/system), and turn embeddings (each word corresponds to a turn number). Cross entropy is used as the loss function.

Given training samples $(\mathcal{C}, \mathcal{B}, \mathcal{A}, \mathcal{R})$, if we train the channel model using complete $(\mathcal{A}, \mathcal{R})$ pairs as input, a significant discrepancy arises between training and decoding for noisy channel online decoding. Since the channel model is used to score partial act and response pairs, that is, $p(\mathcal{C}, \mathcal{B} | \mathcal{O})$ in Algorithm 1, the channel model trained with complete $(\mathcal{A}, \mathcal{R})$ pairs is unsuited to scoring partial sequences. In order to manually create partial sequences during training that are better matched for online decoding, we truncate the $(\mathcal{A}, \mathcal{R})$ pairs with a truncation length uniformly sampled from 1 to the sequence length (inclusive). The direct model and the source model are trained with complete sequences, as partial sequences occur naturally in their standard autoregressive training procedure.

As in-domain dialogue data are usually scarce, we use a two-stage pretraining strategy to enhance the noisy channel model. Although the effectiveness of pretraining with Reddit data has been validated for open-domain dialogue (Zhang et al., 2020b; Bao et al., 2019; Adiwardana et al., 2020), relatively little work has applied such data to task-oriented dialogue.[4] In the first stage, we explore Reddit pretraining (where the Reddit data is pre-processed into $(\mathcal{C}, \mathcal{R})$, i.e., context-response, pairs as described below). In the second stage, we use two task-oriented dialogue datasets, Taskmaster[5] (Byrne et al., 2019) and Schema-Guided Dialogue[6] (Rastogi et al., 2020), to specialize the Reddit-pretrained models. Because the Reddit data consists of open-domain-style dialogues (where belief states and dialogue acts are missing), pretraining on these datasets can familiarize the models with the sequence-to-sequence representation of task-oriented dialogue. Three models, a context-to-response model, a response-to-context model and a response language model, are pretrained to initialize the direct model, the channel model and the source model, respectively.

4.1 Implementation Details

Models: All models are implemented with JAX (Bradbury et al., 2018) and Haiku (Hennigan et al., 2020). For the direct model introduced in Section 2, we use a Transformer model with hidden size 512, 12 encoder-decoder layers, and 16 self-attention heads. The model has 114M parameters. For the noisy channel model, we use a base setting and a large setting. The base setting reduces the number of layers to 5, hidden size to 384, and self-attention heads to 12. Its sub-modules, a direct model, a reverse model and a language model, have 43M, 43M, and 30M parameters, respectively. We employ the base setting for a fair comparison with a single direct model using roughly the same number of parameters (116M vs. 114M). For the large setting, we use the same hyperparameters as the direct model (114M), so that its sub-modules, a direct model, a reverse model, and a language model, have 114M, 114M, and 64M parameters, respectively. We use this large setting to explore the limits of the noisy channel model. The large noisy channel model (292M) is 2.56 times larger compared to the direct model (114M). This illustrates another advantage of the noisy channel model during training. While training a direct model with 292M parameters will overflow the memory of 16GB TPUs (v3)

[4] One exception is Henderson et al. (2019), who use Reddit data to improve response retrieval and selection. We focus on response generation in this work.

[5] https://cutt.ly/xkuUHUa.

[6] https://cutt.ly/QkuUZUu.

Dataset	# Dialog	# Turn	Avg. Turn/Dialog	Avg. Token/Turn	# Domain	Multi-Task	# Unique Slot	# Unique Value
Taskmaster	17,304	341,801	19.75	7.87	7	✗	281	66,659
Schema	22,825	463,284	20.3	9.86	17	✓	123	23,889
CamRest676	676	5,488	8.12	10.71	1	✗	4	89
MultiWOZ	10,438	143,048	13.7	15.03	7	✓	46	11,828
SMCalFlow	41,517	170,590	4.11	8.77	4	✓	–	–

Table 1: Statistics of task-oriented dialogue datasets. We define a multi-task dialogue as a dialogue involving multiple tasks (e.g., hotel and restaurant booking) while its counterpart handles a single task (e.g., hotel booking). Taskmaster and CamRest676 do not contain any multi-task dialogues.

without using model parallelism, training the sub-modules of the large noisy channel model can easily fit into 16GB TPUs, as these modules are independently trained with no need to load three modules for training. This enables us to train a noisy channel model with more parameters compared to training a direct model using the same hardware. For inference, we still need to load the sub-modules into a TPU. Because gradients are not required during inference, we are able to load the three sub-modules of the large noisy channel model (292M) into a single TPU with 16GB memory for decoding. The large noisy channel model (292M) still consumes more memory than the direct model (114M) during inference.

Pretraining Settings: The maximum sequence length l is set to 1024, and sequences with longer lengths are truncated. We reuse the vocabulary from GPT-2 (Radford et al., 2019), which contains 50,257 BPE tokens. We use PreNorm (Nguyen and Salazar, 2019) for faster convergence. GELU (Hendrycks and Gimpel, 2016) is applied as the activation function. Following ALBERT (Lan et al., 2020), dropout is disabled during pretraining. We use the normal distribution truncated to the range $[-0.01, 0.01]$ to initialize the input embeddings, while other parameters are initialized using the normal distribution with zero mean and standard deviation 0.1. The batch size is set to 256. The LAMB optimizer (You et al., 2020) ($b_1 = 0.9$ and $b_2 = 0.999$) is employed for optimization. The initial learning rate is 1e-7, and we apply 4000 warmup steps to increase the learning rate to 1e-3, before utilizing cosine annealing to decay the learning rate. Gradient clipping with clipping value 1 is applied to avoid gradient explosion. We use gradient accumulation with accumulation step 20.

Pretraining: For Reddit pretraining, we download a Reddit dump (with Reddit posts ranging from 2005-12 to 2019-09) from PushShift.[7] Since the comments of a Reddit post are organized into a tree, we extract paths from a tree as dialogue turns. The last comment of each comment path is regarded as the response, while the others are used as the dialogue context. We pretrain each model for 400,000 steps, consuming 102,400,000 (400,000 $\times$ 256) comment paths in total. For the task-oriented pretraining, we combine the two datasets, Taskmaster and Schema-Guided Dialogue, and pretrain for 1e5 steps. The statistics of the task-oriented dialogue datasets are shown in Table 1.

We train each model using 64 TPU chips with 16GB memory each. The pretraining takes around 4 days to complete.

5 Experiments

We fine-tune and evaluate the pretrained models on three dialogue datasets: MultiWOZ 2.0, CamRest676 and SMCalFlow (Andreas et al., 2020). In this section we describe the datasets (Section 5.1), fine-tuning (Section 5.2), decoding (Section 5.3), and evaluation metrics (Section 5.4). Results are presented in Section 6, and analysis and ablation studies in Section 7.

5.1 Datasets

MultiWOZ[8] is a multi-domain dataset consisting of dialogues annotated with $\mathcal{C}, \mathcal{B}, \mathcal{A}, \mathcal{R}$ in the following seven domains: attraction, hotel, hospital, police, restaurant, train, and taxi. Since its release, MultiWOZ has been one of the most commonly used task-oriented dialogue datasets.

CamRest676[9] is annotated similarly to MultiWOZ and consists of dialogues in a single domain: restaurant reservations. Though CamRest676 is

[7] https://pushshift.io/.
[8] https://cutt.ly/0kuUCRS.
[9] https://cutt.ly/SkuUNfE.

smaller than MultiWOZ and predates it, it still provides a widely used benchmark for evaluating task-oriented dialogue models.

SMCalFlow consists of dialogues in four domains: calendar, weather, places, and people. Unlike MultiWOZ and CamRest676, SMCalFlow uses dataflow graphs instead of slot-value pairs to represent belief states and does not annotate dialogue acts. We refer readers to Andreas et al. (2020) for a detailed description of the dataflow representation. We follow Andreas et al. (2020) to convert dataflow graphs into sequences to apply seq2seq models. This dataset is newer and offers fewer prior models to compare with, but we use this dataset to study the robustness of the noisy channel model under different annotation schemas.

We use the public splits for these datasets, where MultiWOZ, CamRest676 and SMCalFlow are split to 8438/1000/1000, 404/136/136, and 32647/3649/5211 dialogues for training, development, and testing, respectively. However, because SMCalFlow's test set has not been publicly released, we randomly select 500 dialogues from its training set to tune hyperparameters and use its development set for testing.

Preprocessing: We use the standard preprocessing procedures for each dataset in order to facilitate fair comparison with previous methods.[10,11,12] In particular, for MultiWOZ and CamRest676, delexicalization is used to reduce lexical variation, while SMCalFlow does not use delexicalization. During delexicalization, slot values are replaced by generic placeholders based on a predefined dictionary. During decoding, following prior work, our dialogue models generate delexicalized responses. These delexicalized responses are re-lexicalized in post-processing by replacing placeholders with their corresponding slot values based on belief states and database information. Since there is no public code for lexicalization,[13] we implement our own functions for lexicalization with regular expressions, for the purpose of displaying example responses. However, this does not affect reported results, as the standard metrics for MultiWOZ and CamRest676 that we adopt here are calculated using delexicalized responses.

[10] https://cutt.ly/TkuU1oM.

[11] https://cutt.ly/zkuU0Ht.

[12] https://cutt.ly/vkuU9bT.

[13] We confirmed this with the dataset authors by email.

5.2 Fine-Tuning

We apply label smoothing with parameter 0.1. Dropout is used on input embeddings and hidden representations, with dropout rate 0.1. The Adam optimizer (Kingma and Ba, 2015) ($b_1 = 0.9$ and $b_2 = 0.999$) is adopted. We use a fixed learning rate 1e-4 with gradient clipping for fine-tuning.

5.3 Decoding

We use direct decoding for belief state. For dialogue act and response, we study three decoding methods: direct decoding, noisy channel reranking, and noisy channel online decoding. Since all of these decoding methods require choosing k_1 tokens from a categorical distribution during expansion, we compare four methods, top-k maximization, sampling without replacement, top-k sampling, and nucleus sampling. Nucleus sampling with cumulative probability 0.98 performs marginally better and is adopted. We perform a range search with the range $[1, 20]$ on development sets for the beam sizes k_1 and k_2, and we set $k_1, k_2 = 4$, $k_1, k_2 = 15$, and $k_1, k_2 = 4$ for MultiWOZ, CamRest676, and SMCalFlow, respectively. For noisy channel reranking and noisy channel online decoding, a grid search with range $[0, 2]$ is performed for λ_1, λ_2, and λ_3. We set ($\lambda_1 = 0.8$, $\lambda_2 = 1$, $\lambda_3 = 0.8$), ($\lambda_1 = 1.2$, $\lambda_2 = 1.2$, $\lambda_3 = 0.8$), and ($\lambda_1 = 0.4$, $\lambda_2 = 1$, $\lambda_3 = 0.2$) for MultiWOZ, CamRest676, and SMCalFlow, respectively.

5.4 Evaluation Metrics

For MultiWOZ and CamRest676, following previous work, we adopt three automatic evaluation metrics: inform, success, and BLEU score. Peng et al. (2020a) showed that these metrics are well correlated to human evaluation. The evaluators[14],[15] provided with the datasets are used for calculating these metrics. To calculate the inform score for a dialogue, the evaluator first checks whether certain placeholders (e.g., `[restaurant_name]`) appear in decoded responses. If so, decoded belief states are converted to database queries to retrieve database records. These database records are compared with the records retrieved with ground-truth belief states. The inform score is one if these two sets of database records match. The success score takes

[14] https://cutt.ly/VkuU3FA.

[15] https://cutt.ly/MkuU88u.

Model	Inform ↑	Success ↑	BLEU ↑	Combined ↑
Sequicity (Lei et al., 2018)	66.4	45.3	15.54	71.39
HRED-TS (Peng et al., 2019)	70.0	58.0	17.50	81.50
DSTC8 Track 1 Winner (Ham et al., 2020)	73.0	62.4	16.00	83.50
DAMD (Zhang et al., 2020a)	76.4	60.4	16.60	85.00
SimpleTOD (Hosseini-Asl et al., 2020)	84.4	70.1	15.01	92.26
SOLOIST (Peng et al., 2020a)	85.5	72.9	16.54	95.74
UBAR (Yang et al., 2021)†	**88.2**	**79.5**	16.43	100.28
Randomly Initialized				
Direct decoding (114M)	81.0	54.7	15.12	82.97
Noisy channel reranking (116M)	82.7	57.1	15.29	85.19
Noisy channel online decoding (116M)	82.9	58.9	15.33	86.23
Noisy channel reranking (292M)	82.1	58.1	15.37	85.47
Noisy channel online decoding (292M)	**83.9**	**60.9**	**15.57**	**87.97**
Reddit Pretraining				
Direct decoding (114M)	81.0	69.2	17.06	92.16
Noisy channel reranking (116M)	81.3	70.1	19.01	94.71
Noisy channel online decoding (116M)	81.6	71.1	19.31	95.66
Noisy channel reranking (292M)	82.2	70.9	19.89	96.44
Noisy channel online decoding (292M)	**82.4**	**71.7**	**20.49**	**97.54**
Task-Oriented Pretraining				
Direct decoding (114M)	85.2	72.9	17.00	96.05
Noisy channel reranking (116M)	85.6	73.8	19.38	99.08
Noisy channel online decoding (116M)	85.9	74.8	19.76	100.11
Noisy channel reranking (292M)	86.5	74.9	20.31	101.01
Noisy channel online decoding (292M)	**86.9**	**76.2**	**20.58**	**102.13**

Table 2: MultiWOZ test results (end-to-end modeling with generated beliefs) with seq2seq approaches. Results are significant ($p < 0.01$) comparing noisy channel decoding and direct decoding. † Yang et al. (2021) also report a combined score of 105.1 with an alternative context and evaluation setting, contributions orthogonal to our work and the other benchmarks reported here.

all the requestable slots (e.g., postcode, phone number, and address) from a decoded response and compares these requestable slots with the ones in the ground-truth response. The success score is one if generated requestable slots coincide with the ground-truth ones. BLEU score (BLEU-4) compares the n-grams of generated responses and human responses, and is a widely used metric in NLP for evaluating text quality. Following Budzianowski et al. (2018), we also calculate a combined score, which is (Inform + Success) / 2 + BLEU. For SMCalFlow, inform and success scores are not applicable because calculation of these scores relies on delexicalization placeholders, and this dataset does not use delexicalization. We use SacreBLEU[16] and TER[17] to directly measure the quality of responses. As prior work on this dataset has focused on belief tracking rather than end-to-end response generation, we are the first to use these metrics on this dataset.

We perform significance tests, where we use t-test for inform, success, and TER scores and use permutation test for BLEU.

6 Results

MultiWOZ: Results on the MultiWOZ test set are shown in Table 2. We observe several trends. First, the base noisy channel model (116M) performs better than direct decoding (114M), despite having a similar number of parameters, showing that the noisy channel factorization is beneficial for task-oriented dialogue. The large noisy channel setting improves further over the base setting. Second, Reddit pretraining provides benefits over random initialization, validating the use of large

[16] https://cutt.ly/BkuU7dL.

[17] https://pypi.org/project/pyter/.

Model	Inform ↑	Success ↑	BLEU ↑	Combined ↑
Sequicity (Lei et al., 2018)	92.3	85.3	21.40	110.20
GPT-2 fine-tuned (Wu et al., 2019b)	-	86.2	19.20	-
ARDM (Wu et al., 2019b)	-	**87.1**	25.20	-
SOLOIST (Peng et al., 2020a)	94.7	**87.1**	25.50	116.40
Randomly Initialized				
Direct decoding (114M)	78.1	83.5	21.58	102.38
Noisy channel online decoding (116M)	79.8	84.1	22.83	104.78
Noisy channel online decoding (292M)	**80.9**	**84.9**	**23.19**	**106.09**
Reddit Pretraining				
Direct decoding (114M)	93.3	83.9	23.41	112.01
Noisy channel online decoding (116M)	93.7	84.5	25.14	114.24
Noisy channel online decoding (292M)	**93.9**	**84.7**	**25.38**	**114.68**
Task-Oriented Pretraining				
Direct decoding (114M)	93.4	84.3	24.92	113.77
Noisy channel online decoding (116M)	94.3	85.2	25.98	115.73
Noisy channel online decoding (292M)	**95.4**	**85.3**	**26.89**	**117.24**

Table 3: CamRest676 test results (end-to-end modeling with generated beliefs) with seq2seq approaches. Noisy channel reranking performs comparable with noisy channel online decoding, and the results are not shown. Results are significant ($p < 0.01$) comparing noisy channel decoding and direct decoding.

Model	SacreBLEU ↑	TER ↓
Randomly Initialized		
Direct decoding (114M)	51.30	89.13
Online decoding (116M)	53.66	74.18
Online decoding (292M)	**54.39**	**73.18**
Reddit Pretraining		
Direct decoding (114M)	60.68	61.99
Online decoding (116M)	63.29	47.16
Online decoding (292M)	**63.91**	**46.43**
Task-Oriented Pretraining		
Direct decoding (114M)	61.02	59.84
Online decoding (116M)	63.72	46.27
Online decoding (292M)	**64.29**	**45.81**

Table 4: SMCalFlow results. Reranking performs worse than online decoding, and the results are not shown. Results are significant ($p < 0.01$) comparing noisy channel decoding and direct decoding.

open-domain dialogue-genre pretraining for task-oriented dialogue, while the models with a second stage of task-oriented pretraining obtain further improvements. This effect is consistent across both direct and noisy channel decoding. Finally, we observe that online decoding consistently outperforms reranking, indicating the benefits of tighter model integration during decoding.

Our model performs better on combined score than SOLOIST (Peng et al., 2020a), a closely related baseline that pretrains a GPT2-initialized Transformer with Taskmaster and Schema-Guided Dialogue and decodes with nucleus sampling.

CamRest676: Results on the CamRest676 test set are shown in Table 3. We observe that the base noisy channel model (116M) obtains better results compared to direct decoding (114M), again demonstrating the effectiveness of the noisy channel model. Reddit pretraining again provides a large benefit over random initialization for both direct decoding and noisy channel decoding, while task-oriented pretraining provides a further boost. Our model again performs better than SOLOIST.

SMCalFlow: Results on the SMCalFlow development set are shown in Table 4. As end-to-end models have not previously been tested on this dataset, we use it to demonstrate that the noisy channel model, which we developed primarily on MultiWOZ, continues to be effective on task-oriented dialogue datasets with different annotation schema. The results are consistent with MultiWOZ and CamRest676. The noisy channel model outperforms the direct model by a large margin, demonstrating that dialogue act annotations are not essential for the noisy channel

Model	CamRest676	MultiWOZ
Direct decoding	115.17	96.73
Noisy Channel Online Decoding		
Direct + Channel	115.63	98.54
Direct + Source	115.91	99.12
Direct + Length	115.56	97.57
Channel + Source	115.82	99.18
Channel + Length	115.60	98.71
Source + Length	115.62	99.19
All - Direct	115.96	100.89
All - Channel	116.56	100.93
All - Source	116.38	99.92
All - Length	116.52	101.11
All	**116.91**	**102.62**

Table 5: Ablation results for model combination on development sets (combined score). Results for reranking are similar and are not shown. 'All', 'Direct', 'Source', and 'Channel' denote no ablation, direct model, source model and channel model, respectively. Rows with '+' are combinations of two sub-modules, while the rows with '-' are combinations of three sub-modules.

model, and that it remains effective across diverse dialogue representations.

Reddit pretraining confers a similar large benefit on SMCalFlow as on the other datasets, but we observe that task-oriented pretraining brings only marginal further improvements. This may be due to differences in domain or format between our pretraining datasets and SMCalFlow. Alternatively, task-oriented pretraining may help more on task-specific metrics, such as inform and success scores, than on text quality metrics such as BLEU and TER scores. This hypothesis is further supported by the MultiWOZ results in Table 2.

7 Analysis

In this section, we use MultiWOZ and CamRest676 to perform ablation studies on the effects of model combination, large-scale pretraining, and sample efficiency; as well as analyzing the runtime requirements of our model and the reasons for its success.

7.1 Ablation on Model Combination

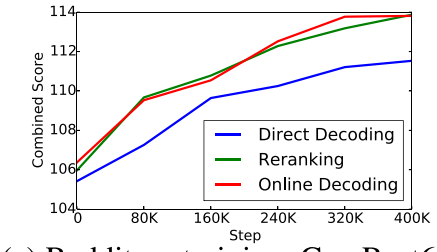
Noisy channel decoding involves a combination of four sub-modules, as in Eq. 1: the direct model, channel model, language model, and length bias. We perform an ablation study to determine whether all model components are important to the result, using the large model. Results on the development sets of CamRest676 and MultiWOZ are presented in Table 5. Note that the ablation is performed after applying the direct model to obtain k_1 expansions at each beam search step for noisy channel online decoding. We find that the combination of all four sub-modules performs the best, followed by combinations of three and then two sub-modules. The results are significant when comparing 'All' and the baselines ($p < 0.01$). This result demonstrates the effectiveness of the noisy channel factorization, and the importance of each model component.

7.2 Effect of Pretraining Scale

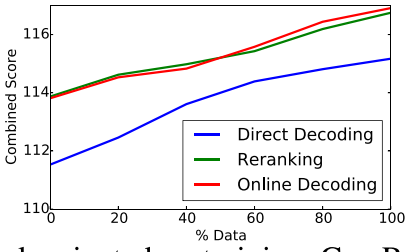
We investigate the importance of scale for both our pretraining stages. We select different checkpoints for Reddit pretraining, and truncate the two task-oriented dialogue datasets for task-oriented pretraining. We fine-tune these models using the full training data of CamRest676 or MultiWOZ. The results of three decoding methods (with the large noisy channel model) on the development sets are shown in Figure 2. In Figure 2 (a) and (c), the combined scores of all three decoding methods improve with more Reddit pretraining steps, demonstrating the advantage of increasing amounts of data in the open-domain dialogue pretraining stage. In Figure 2 (b) and (d), the combined scores further increase with more task-oriented data, confirming that additional task-oriented pretraining data is useful.

7.3 Sample Efficiency of Fine-Tuning

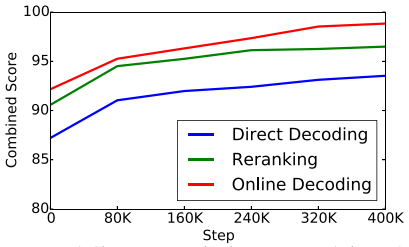
We investigate whether pretraining can improve sample efficiency during fine-tuning. We gradually increase the amount of fine-tuning data and evaluate the randomly-initialized, Reddit pretrained and task-oriented pretrained models. The results on the development sets are shown in Figure 3. Combined scores increase with more training data under all conditions. Crucially, Reddit pretrained models show better performance with a smaller amount of fine-tuning data than randomly initialized models, and task-oriented pretrained models better still. We conclude that both our pretraining stages can improve sample efficiency, which is especially important when the target task has little training data.



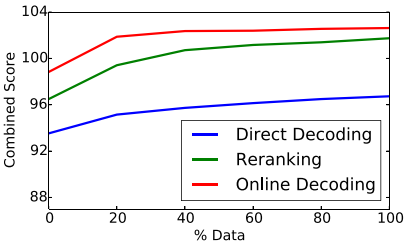
(a) Reddit pretraining, CamRest676



(b) Task-oriented pretraining, CamRest676

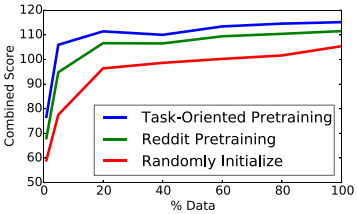


(c) Reddit pretraining, MultiWOZ

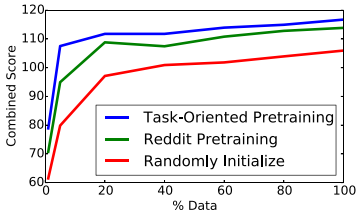


(d) Task-oriented pretraining, MultiWOZ

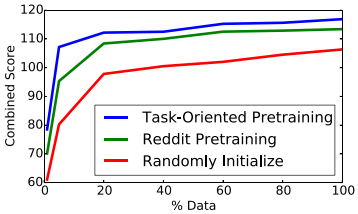
Figure 2: Results showing the effect of pretraining scale.



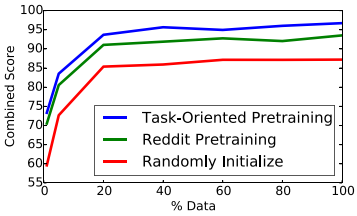
(a) Direct decoding, CamRest676



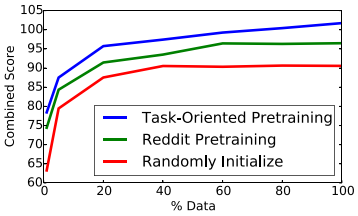
(b) Reranking, CamRest676



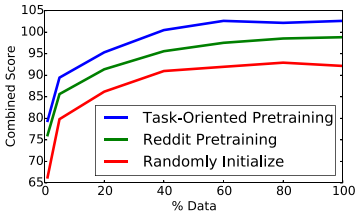
(c) Online decoding, CamRest676



(d) Direct decoding, MultiWOZ



(e) Reranking, MultiWOZ



(f) Online decoding, MultiWOZ

Figure 3: Pretraining improves sample efficiency during fine-tuning.

7.4 Decoding Runtime

In Table 6, we report the average clock time for decoding one turn (including its belief state, dialogue act and response). Noisy channel reranking is slightly slower compared to direct decoding, with overhead due to the reranking step in Eq. 1. Noisy channel online decoding is significantly slower, since it needs to apply Eq. 1 at each beam search step. In future work we will investigate ways to improve the efficiency of online decoding.

7.5 Decoding Properties

In this section we analyze why the noisy channel model performed better than direct decoding.

Model	CamRest676	MultiWOZ
Direct decoding	**4.89**	**6.48**
Reranking	5.43	6.92
Online decoding	8.73	10.97

Table 6: Average decoding time (in seconds) for each turn with different decoding methods.

Length: In Table 7 we show the average length of generated responses. Direct decoding produces shorter responses than the ground truth, confirming that the direct model prefers short and generic responses. Adding a length bias to direct decoding (with lambda tuned on the development sets) produces responses longer than the ground truth,

Model	CamRest676	MultiWOZ
Ground truth	14.50	16.91
Direct decoding	12.07	12.85
Direct decoding + Length	15.98	17.73
Reranking	**15.09**	17.47
Online decoding	15.14	**17.32**

Table 7: The average length of responses with different decoding methods (on test set). The value closest to the ground truth is bold.

Model	CamRest676	MultiWOZ
Ground truth	1.07	1.22
Direct decoding	0.84	0.91
Reranking	0.87	0.99
Online decoding	**0.89**	**1.03**

Table 8: The Zipf scores of responses with different decoding methods (on test set). The value closest to the ground truth is bold.

Model	CamRest676	MultiWOZ
Direct decoding	0.24	0.31
Reranking	0.12	0.14
Online decoding	**0.08**	**0.11**

Table 9: The likelihood (%) of falling into repetition loops for different decoding methods (on test set).

which may be a disadvantage. The noisy channel models produce responses with average length closest to the ground truth.

Zipf: Table 8 shows the Zipf scores of responses. We find that the word distributions of responses generated by the noisy channel models are closer to the word distribution of ground-truth responses.

Repetition: In Table 9 we examine the likelihood of falling into repetition loops (Holtzman et al., 2019) for different decoding methods. Repetition loops are rare for all decoding methods, but noisy channel decoding can further decrease their likelihood. The channel model can discount a sequence with a repetition loop, since it conveys less information than a natural sequence of the same length, making it harder to ‘‘explain’’ the context.

Examples: Some examples of responses are shown in Table 10. We observe that noisy channel models decode longer responses compared to direct decoding, and that the responses can explain their dialogue contexts well to meet users' requirements.

8 Related Work

Task-Oriented Dialogue Models: Most task-oriented dialogue systems break down the task into three components: belief tracking (Henderson et al., 2013; Mrkšić et al., 2016; Rastogi et al., 2017; Nouri and Hosseini-Asl, 2018; Wu et al., 2019a; Zhang et al., 2019; Zhou and Small, 2019; Heck et al., 2020), dialogue act prediction (Wen et al., 2017a; Tanaka et al., 2019), and response generation (Chen et al., 2019; Budzianowski et al., 2018; Lippe et al., 2020). Traditionally, a modular approach is adopted, where these components are optimized independently (i.e., a pipeline design) or learned via multi-task learning (i.e., some parameters are shared among the components) (Wen et al., 2017b; Neelakantan et al., 2019; Zhao et al., 2019; Mehri et al., 2019; Tseng et al., 2020; Lee et al., 2020). However, it is known that improvements in one component do not necessarily lead to overall performance improvements (Ham et al., 2020), and the modular approach suffers from error propagation in practice (Liu and Lane, 2018). These observations gave rise to the sequence-to-sequence approach (Lei et al., 2018; Pei et al., 2019; Budzianowski and Vulić 2019; Wu et al., 2019b; Zhang et al., 2020a; Ham et al., 2020; Hosseini-Asl et al., 2020; Peng et al., 2020a; Yang et al., 2021), where dialogue beliefs and acts are represented as text spans, and a sequence-to-sequence model is applied to subsume the three components. Our work is situated within this general approach. In contrast to previous work, however, which uses a direct model for decoding, we introduce the noisy channel model to improve task-oriented dialogue.

Pretraining Models for Dialogue: Recent work has applied pretraining (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019) to dialogue. For open-domain dialogue, DialoGPT (Zhang et al., 2020b) and CGRG (Wu et al., 2020b) extend GPT-2 (Radford et al., 2019) for response generation. PLATO (Bao et al., 2019) and PLATO-2 (Bao et al., 2020) pretrain a latent variable model with social media data for diversified response generation. Meena (Adiwardana et al., 2020) collects a large-scale social media corpus

Model	Delexicalized Response	SacreBLEU ↑
	Dialogue: MUL1624, Turn Number: 4	
Ground truth	Sure. The trainid is [train_id]. It leaves at [value_time] and arrives at [value_time]. Would you like to make a reservation?	–
Direct decoding	The travel time is [value_count] minutes.	2.03
Reranking	[train_id] leaves [value_place] at [value_time] and arrives in [value_place] at [value_time]. Would you like me to book that for you?	46.33
Online decoding	[train_id] leaves at [value_time] and arrives at [value_time]. Would you like me to book that?	**60.90**
	Dialogue: MUL1276, Turn Number: 6	
Ground truth	The [restaurant_name] has a postal code of [restaurant_postcode]. Will this work for you? I can book it if you please.	–
Direct decoding	There are [value_count] places. What type of food would you like?	1.63
Reranking	How about [restaurant_name] located at [restaurant_address], postcode [restaurant_postcode].	**24.90**
Online decoding	How about [restaurant_name] located at [restaurant_address], postcode [restaurant_postcode]?	22.00
	Dialogue: MUL1898, Turn Number: 2	
Ground truth	Yes, [attraction_name] is on [attraction_address] and is in the [value_area] side of town. Is there anything else you need to know?	–
Direct decoding	[attraction_name] is located in the [value_area] part of town and has free admission.	27.53
Reranking	[attraction_name] is located in the [value_area] of town at [attraction_address]. The entrance fee is free. Can I help you with anything else?	41.66
Online decoding	[attraction_name] is located in the [value_area] part of town at [attraction_address]. Can I help you with anything else?	**42.38**

Table 10: Case study on the responses decoded by direct decoding, noisy channel reranking, and noisy channel online decoding. The large noisy channel model is used.

for pretraining and proposes a metric named sensibleness and specificity average for evaluation. Roller et al. (2020) study various strategies for building an open-domain chatbot with Reddit for pretraining. For task-oriented dialogue, ToD-BERT (Wu et al., 2020a) fine-tunes BERT (Devlin et al., 2019) for four tasks, including intention detection, belief tracking, dialogue act prediction, and response selection. SC-GPT (Peng et al., 2020b) fine-tunes GPT-2 for few-shot response generation with given dialogue acts. Ham et al. (2020) fine-tune GPT-2 for belief tracking and context-to-response generation. SimpleTOD (Hosseini-Asl et al., 2020) proposes a method to serialize dialogue beliefs and acts into text spans and fine-tunes GPT-2 for end-to-end dialogue modeling. SOLOIST (Peng et al., 2020a) uses a series of task-oriented dialogue datasets to further pretrain GPT-2 before fine-tuning it on final tasks for evaluation. Unlike these BERT- or GPT-initialized task-oriented dialogue models, which are essentially pretrained with general text, such as Wikipedia and BookCorpus, we use a Reddit dump to pretrain the models to learn from open-domain dialogues.

9 Conclusion

We introduced two noisy channel models, noisy channel reranking and noisy channel online decoding, for task-oriented dialogue. Large-scale pretraining was further adopted to tackle data scarcity in downstream tasks. Extensive experiments on MultiWOZ, CamRest676, and SMCalFlow demonstrated that (1) the noisy channel models significantly outperform direct decoding; (2) models with pretraining improve over randomly-initialized models; (3) the models are robust to different dialogue schema annotations; and (4) the noisy channel models can decode responses closer to ground-truth responses than direct decoding.

Acknowledgments

We would like to thank the action editors (Maggie, Wenjie Li, and Eneko Agirre) and three anonymous reviewers for their insightful comments. We also thank Angeliki Lazaridou, Gábor Melis, Nando de Freitas, Chris Dyer, and the DeepMind language team for their helpful discussions.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, and Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

John Langshaw Austin. 1975. *How To Do Things with Words*, 88, Oxford University Press.

Siqi Bao, Huang He, Fan Wang, and Hua Wu. 2019. Plato: Pre-trained dialogue generation model with discrete latent variable. *arXiv preprint arXiv:1910.07931*.

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: pre-trained dialogue generation model with discrete latent variable. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 85–96. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.9

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. 2018. JAX: composable transformations of Python+ NumPy programs.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, Prague, Czech Republic. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutsk. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's GPT-2 - how can I help you? Towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22, Hong Kong. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-5602

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1547

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 2019, Hong Kong, China, November 3–7, 2019*, pages 4515–4524. Association for Computational Linguistics.

https://doi.org/10.18653/v1/D19-1459

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1360

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. Iryna Gurevych and Yusuke Miyao, editors, In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

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. Association for Computational Linguistics, Online.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2020, 1st virtual meeting, July 1-3, 2020*, pages 35–44. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Steve Young. 2013. Deep neural network approach for the dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 467–471.

Matthew Henderson, Ivan Vulic, Daniela Gerz, Iñigo Casanueva, Pawel Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrksic, and Pei-Hao Su. 2019. Training neural response selection for task-oriented dialogue systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5392–5404. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1536

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Tom Hennigan, Trevor Cai, Tamara Norman, and Igor Babuschkin. 2020. Haiku: Sonnet for JAX.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

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*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.

Dan Klein and Christopher D. Manning. 2002. Conditional structure versus conditional estimation in NLP models. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 9–16. https://doi.org/10.3115/1118693.1118695

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*.

Hwaran Lee, Seokhwan Jo, HyungJun Kim, Sangkeun Jung, and Tae-Yoon Kim. 2020. Sumbt+ larl: End-to-end neural task-oriented dialog system with reinforcement learning. *arXiv preprint arXiv:2009.10447*.

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.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016*, pages 110–119. The Association for Computational Linguistics.

Phillip Lippe, Pengjie Ren, Hinda Haned, Bart Voorn, and Maarten de Rijke. 2020. Diversifying task-oriented dialogue response generation with prototype guided paraphrasing. *CoRR*, abs/2008.03391.

Bing Liu and Ian Lane. 2018. End-to-end learning of task-oriented dialogs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 67–73. https://doi.org/10.18653/v1/N18-4010

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. *arXiv preprint arXiv:1907.10016*.

Nikola Mrkšić, Diarmuid O. Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2016. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*.

Arvind Neelakantan, Semih Yavuz, Sharan Narang, Vishaal Prasad, Ben Goodrich, Daniel Duckworth, Chinnadhurai Sankar, and Xifeng Yan. 2019. Neural assistant: Joint action prediction, response generation, and latent knowledge reasoning. *arXiv preprint arXiv:1910.14613*.

Toan Q. Nguyen and Julian Salazar. 2019. Transformers without tears: Improving the normalization of self-attention. *arXiv preprint arXiv:1910.05895*.

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking model. *arXiv preprint arXiv:1812.00899*.

Jiahuan Pei, Pengjie Ren, and Maarten de Rijke. 2019. A modular task-oriented dialogue system using a neural mixture-of-experts. *arXiv preprint arXiv:1907.05346*.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, 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, Xiujun 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, EMNLP 2020, Online Event, 16–20 November 2020*, pages 172–182. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.17

Shuke Peng, Xinjing Huang, Zehao Lin, Feng Ji, Haiqing Chen, and Yin Zhang. 2019. Teacher-student framework enhanced multi-domain dialogue generation. *arXiv preprint arXiv:1908.07137*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, pages 2227–2237. https://doi.org/10.18653/v1/N18-1202

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 561–568. IEEE. https://doi.org/10.1109/ASRU.2017.8268986

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, February 7–12, 2020*, pages 8689–8696. AAAI Press. https://doi.org/10.1609/aaai.v34i05.6394

Antoine Raux, Brian Langner, Dan Bohus, Alan W Black, and Maxine Eskenazi. 2005. Let's go public! taking a spoken dialog system to the real world. In *Ninth European conference on speech communication and technology*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, Y-Lan Boureau, and Jason Weston. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Stephanie Seneff and Joseph Polifroni. 2000. Dialogue management in the Mercury flight reservation system. In *ANLP-NAACL 2000 Workshop: Conversational Systems*. https://doi.org/10.3115/1117562.1117565

Claude Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423.

Koji Tanaka, Junya Takayama, and Yuki Arase. 2019. Dialogue-act prediction of future responses based on conversation history. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 197–202.

Bo-Hsiang Tseng, Jianpeng Cheng, Yimai Fang, and David Vandyke. 2020. A generative model for joint natural language understanding and generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 1795–1807. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.163

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve J. Young. 2017a. Latent intention dialogue models. *CoRR*, abs/1705.10229.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina Maria Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve J. Young. 2017b. 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, EACL 2017, Valencia, Spain, April 3–7, 2017, Volume 1: Long Papers*, pages 438–449. Association for Computational Linguistics.

Chien-Sheng Wu, Steven C. H. Hoi, Richard Socher, and Caiming Xiong. 2020a. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16–20, 2020*, pages 917–929. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019a. Transferable multi-domain state generator for task-oriented dia-

logue systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 808–819. Association for Computational Linguistics.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2019b. Alternating recurrent dialog model with large-scale pre-trained language models. *arXiv preprint arXiv:1910.03756*.

Zeqiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, and Bill Dolan. 2020b. A controllable model of grounded response generation. *arXiv preprint arXiv:2005.00613*.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. Ubar: Towards fully end-to-end task-oriented dialog systems with gpt-2. *The Thirty-Fifth AAAI Conference on Artificial Intelligence*.

Kyra Yee, Yann N. Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 5695–5700. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1571

Yang You, Jing Li, Sashank J. Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training BERT in 76 minutes. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*.

Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomás Kociský. 2017. The neural noisy channel. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*.

Lei Yu, Laurent Sartran, Wojciech Stokowiec, Wang Ling, Lingpeng Kong, Phil Blunsom, and Chris Dyer. 2020. Better document-level machine translation with bayes rule. *Transactions of the Association for Computational Linguistics*, 8:346–360. https://doi.org/10.1162/tacl_a_00319

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S. Yu, Richard Socher, and Caiming Xiong. 2019. Find or classify? Dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv preprint arXiv:1910.03544*.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020a. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020*, pages 9604–9611. AAAI Press. https://doi.org/10.1609/aaai.v34i05.6507

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. 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, ACL 2020, Online, July 5–10, 2020*, pages 270–278. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-demos.30

Tiancheng Zhao, Kaige Xie, and Maxine Eskénazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers)*, pages 1208–1218. Association for Computational Linguistics.

Li Zhou and Kevin Small. 2019. Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering. *arXiv preprint arXiv:1911.06192*.