Alternating Recurrent Dialog Model with Large-scale Pre-trained Language Models

Qingyang Wu[1]**, Yichi Zhang**[2]**, Yu Li**[1]**, Zhou Yu**[1]
[1]University of California, Davis, [2]Tsinghua University,
{wilwu, yooli, joyu}@ucdavis.edu, zhangyic17@mails.tsinghua.edu.cn

Abstract

Existing dialog system models require extensive human annotations and are difficult to generalize to different tasks. The recent success of large pre-trained language models has suggested the effectiveness of incorporating language priors in down-stream NLP tasks. However, how much pre-trained language models can help dialog response generation is still under exploration. In this paper, we propose a simple, general, and effective framework: Alternating Recurrent Dialog Model (ARDM)[1]. ARDM models each speaker separately and takes advantage of large pre-trained language models. It requires no supervision from human annotations such as belief states or dialog acts to achieve effective conversations. ARDM outperforms or is on par with the state-of-the-art methods on two popular task-oriented dialog datasets: CamRest676 and MultiWOZ. Moreover, we can generalize ARDM to more challenging, non-collaborative tasks such as persuasion. In the PersuasionForGood task, ARDM is capable of generating human-like responses to persuade people to donate to a charity.

1 Introduction

It has been a long-standing ambition for artificial intelligence researchers to create an intelligent conversational agent that can generate human-like responses. Recently, data-driven dialog models are more and more popular. However, most current state-of-the-art approaches still heavily rely on extensive human annotations such as belief states and dialog acts (Lei et al., 2018). However, dialog content can vary considerably in different dialog tasks. Having a different intent or dialog act annotation scheme for each task is costly and even impossible for tasks such as open-domain social chat. Thus, it is difficult to utilize these methods on challenging dialog tasks where dialog states and acts are difficult to annotate such as persuasion and negotiation.

Eric and Manning (2017) proposed a simple sequence-to-sequence architecture that requires no explicit annotations. The model learns to extract information from dialog history with attention and copy mechanism. However, due to the limited language modeling capability in the previous model, Sequicity (Lei et al., 2018), which uses belief states as inputs for supervision, outperforms Eric and Manning (2017)'s method significantly in the recent dialog datasets. But with the success of large pre-trained language models such as BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), we investigate how large-scale pre-trained language models can help dialog tasks.

Previous sequence-to-sequence models are used to tackle documents with only one narrator. However, in dialogs, two speakers have different roles; therefore, their language model distributions are very different from each other. To address this issue, we propose ARDM, a dialog model that encodes and decodes different speaker utterances in alternating order. This structure makes the model more flexible and efficient than traditional sequence-to-sequence models in processing various dialogs. We evaluate our model on three different task-oriented dialog datasets: CamRes676, MultiWOZ, and PersuasionForGood. The first two datasets are traditional information request dialog datasets with well-defined automatic evaluation metrics on task completion. By contrast, PersuasionForGood is a new dataset that focuses on persuading people to donate to a charity. Due to the complexity of dialog content, there is no explicit dialog state defined in this task.

We observe that ARDM is capable of improving task-oriented dialog tasks performance over the previous state-of-the-art methods without in-

[1] https://github.com/qywu/ARDM

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1292–1301
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

corporating any explicit supervision from belief states or dialog acts. Also, because of ARDM's simplicity and generality, one can rapidly build a dialog prototype on different types of applications using only conversations without additional human annotations. We also found that ARDM works well on complex dialogs, such as persuasion. The model generates dialog responses that successfully persuade people to donate to a charity, suggesting the potential of ARDM being used in wide-scale real-world settings.

2 Related Work

Traditional dialog systems consist of a dialog manager to maintain dialog states and control the conversation flow. However, a dialog manager requires extensive manual annotations for training the sub-modules such as dialog state tracker or policy decision-maker. An alternative is to model dialog without explicitly modeling belief states. Specifically, Eric and Manning (2017) proposed a sequence-to-sequence model that utilizes copy-mechanism to copy history information directly from raw dialog history. This method achieved the state-of-the-art results on DSTC2 (Henderson et al., 2014), which is a simple dialog restaurant booking task with abundant data. However, such method did not perform well on more complex dialog task data sets CamRes676 (Wen et al., 2017) and KVRET (Eric et al., 2017). Sequicity (Lei et al., 2018) attributed the bad performance of Eric and Manning (2017)'s method to the omission of belief tracker. They introduced the concept of belief span and added belief tracker back to the model and achieved state-of-the-art performance.

Compared to Sequicity, Eric and Manning (2017)'s method provides a more general framework that reduces manual dialog state, user intent, and dialog act labeling by bypassing any symbolic annotations. Such a model can apply to datasets with no or partial annotations of belief states. In a real-world setting, if the dialog task introduces new slot values in belief states (i.e. a new type of food), Sequicity will suffer from the belief span decoder error in response generation. Thus, Eric and Manning (2017)'s method may be potentially more robust than Sequicity in this situation. Besides, if the task requires belief states for database search, we can treat belief tracking as a separate task. We can train a good belief tracking with only a small amount of annotated data, which reduces the annotation required and is easier to fix errors. Also, since belief states are a set of important entities condensed from dialog history (i.e., often exact words from utterances), they do not introduce extra information to the model. Therefore, a dialog model with powerful representation learning should learn a form of belief states information automatically without human annotations as the scaffold.

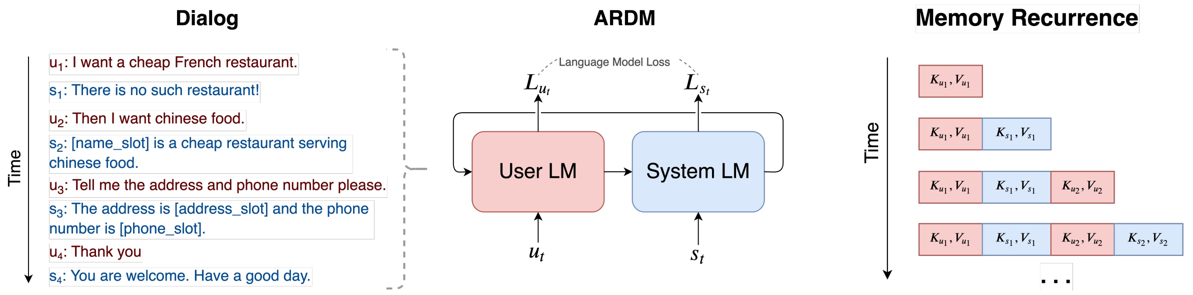
Recent success of BERT (Devlin et al., 2019) and GPT2 (Radford et al., 2019) suggests the possibility of applying large pre-trained language models to dialog systems. There are some studies of applying large pre-trained language model to dialog generation. TransferTransfo (Wolf et al., 2019) fine-tuned the pre-trained language model GPT (Radford et al., 2018) on Persona-Chat dataset (Zhang et al., 2018) and obtained significant improvements on chitchat response generation, suggesting the potential of fine-tuning large pre-trained language models on other dialog response generation tasks. A more recent work (Budzianowski and Vulic, 2019) adopted the framework of TransferTransfo. They made the first attempt to leverage large pre-trained language models GPT and GPT-2 on task-oriented dialog generation, but it included belief states modeling as the input and did not achieve better results than the baseline. We propose to model dialogs without any annotation but rely on pre-trained large-scale language models that alternate.

3 Alternating Recurrent Dialog Model

We propose Alternating Recurrent Dialog Model (ARDM) by compositing two separate pre-trained language model in alternate order to learn the user and system utterance distribution through memory recurrence. Figure 1 shows an overview of ARDM.

3.1 Recurrent Modeling for User and System

We aim to model both user and system utterances distribution recurrently. Given a multi-turn dialog (d) between a user (u) and a system (s), we can represent d as a series of utterances $\{u_1, s_1, u_2, s_2, \ldots, u_T, s_T\}$, where T denotes the total number of turns. We decompose the probability distributions over the utterances in d into two language models for the user and system respectively, denoted as p_u and p_s. Then we define a


Dialog
Time
u1: I want a cheap French restaurant.
s1: There is no such restaurant!
u2: Then I want chinese food.
s2: [name_slot] is a cheap restaurant serving chinese food.
u3: Tell me the address and phone number please.
s3: The address is [address_slot] and the phone number is [phone_slot].
u4: Thank you
s4: You are welcome. Have a good day.
ARDM
Language Model Loss
L_{u_t}
L_{s_t}
User LM
System LM
u_t
s_t
Memory Recurrence
Time
K_{u_1}, V_{u_1}
K_{s_1}, V_{s_1}
K_{u_2}, V_{u_2}
K_{s_2}, V_{s_2}
...

(a) (b)

Figure 1: Alternating Recurrent Dialog Model (ARDM) Overview. (a) shows how we feed the entire dialog to ARDM. (b) shows the recurrence mechanism we used to preserve memory.

dialog model $p(d)$ with the equation:

$$p(d) = \prod_{t=1}^{T} p_u(u_t|u_{<t}, s_{<t})\, p_s(s_t|u_{\leq t}, s_{<t}) \quad (1)$$

p_u and p_s are standard language models where the task is to predict the next token given the preceding context. For an utterance u_t or s_t with m tokens $\{w_1, \ldots, w_m\}$, the joint probability of an utterance is as follows:

$$p_u(u_t|u_{<t}, s_{<t}) = \prod_{i=1}^{m_{u_t}} P(w_i|w_{<i}, u_{<t}, s_{<t}) \quad (2)$$

$$p_s(s_t|u_{\leq t}, s_{<t}) = \prod_{i=1}^{m_{s_t}} P(w_i|w_{<i}, u_{\leq t}, s_{<t}) \quad (3)$$

Finally, we train the dialog model by maximizing the likelihood over Equation 1. However, in order to model multi-turn dialog distributions efficiently, we need a memory mechanism to encode the history. In the next section, we introduce the memory mechanism in detail.

3.2 Memory Recurrence

We apply a memory mechanism to grant the model the capability of memorizing conversation history. This method is similar to Transformer-XL (Dai et al., 2019), which enables the model to learn longer dependency. For an utterance at turn t, we reuse the hidden states $h_{\leq t-1}$ stored in the memory M_{t-1} to obtain h_t, and store the h_t back to the memory as M_t. As for the pre-trained Transformer language model, we implement the memory mechanism using self-attention given the query/key/value features denoted as Q, K, V, where the equation is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T V) \quad (4)$$

For simplicity, we assume there is only one layer in Transformer, and h_t is the hidden states which consist of N vectors for the current input N tokens in the utterance at time t. Then a recurrence relation for h_t is defined by computing Q_t, $K_{\leq t}$, $V_{\leq t}$ from $h_{\leq t-1}$ and the current utterance. In practice, we reuse $K_{\leq t-1}$ and $V_{\leq t-1}$ (i.e. history keys and values) as M_{t-1} instead of h_{t-1} to avoid recomputing history information. Therefore, the final h_t is computed as:

$$\begin{aligned} M_{t-1} &= [K_{\leq t-1}, V_{\leq t-1}] && (5)\\ K_{\leq t}, V_{\leq t} &= [K_{\leq t-1}; K_t], [V_{\leq t-1}; V_t] && (6)\\ h_t &= \text{Attention}(Q_t, K_{\leq t}, V_{\leq t}) && (7) \end{aligned}$$

One can use h_t (consisting of vectors for each token) to get each token's probability to calculate the language model cross entropy loss to maximize $p(w_i|w< i, u_{<t}, s_{<t})$, shown in Figure 1.

3.3 Training Details

We initialize the user and the system language model with a large pre-trained language model GPT-2 small with 117M parameters (Radford et al., 2019). It is a Transformer (Vaswani et al., 2017) model with 12 heads, 768 hidden size, and 12 layers. The model is trained on a large scale corpus called WebText extracted from Reddit with at least three upvotes. The tokenizer is 50,257 size byte pair encoding (BPE) (Sennrich et al., 2016) that can encode and decode any text in a lossless manner to avoid out-of-vocabulary tokens. We follow a special format in GPT-2 as the "trigger" so that the model can zero-shot dialog response generation, by prefixing the user role token "A:" or "B:", and suffixing the end of utterance token “\n\n\n”. This “trigger” approach is similar in other zero-shot scenarios mentioned in GPT-2 paper (e.g., that a ”TL;DR” token can trigger GPT-2 to summarize the input text.) We further fine-tune ARDM on the specific task dataset. We apply AdamW optimizer (Loshchilov and Hutter, 2019), and the number of warm-up steps is set to be the number of batches in one epoch. The learning rate is set to 3×10^{-5}, and the dropout rate is set to 0.1 for all tasks. We conduct all experiments on a 11GB GPU.

3.4 Decoding Details

We decode utterances by nucleus sampling (Holtzman et al., 2019) with different hyper-parameters (top-p, top-k) for down-stream dialog tasks. We also vary the temperature of $T < 1$ to find the best setting for the specific down-stream dialog task. To handle both situations in the evaluation and the real-world use case, we have two decoding modes. For evaluation mode, we feed all past ground truth history before turn t to generate the corresponding utterance, so that we can evaluate the quality of generated dialog responses without concerning about the conversion flow. While in a real-world use case, we do not have ground truth history, and therefore we use the memory from previously generated responses and let the model dynamically interact with a human or another bot in turns. Because dialogs have different lengths, it is hard for ARDM to efficiently decode responses using traditional batch padding method. As a solution, we develop a dynamic dialog filtering algorithm to support fast decoding in batch. Such method speeds up the generation eight times faster. Please refer to Appendix for the method’s details.

3.4.1 Belief Tracking

Database state means the database query results. In the example, since there is no match in the database satisfying the constraint ”cheap french restaurant”, it returns ”0” and the domain ”restaurant”. Then the system can condition on the database state to generate ”There is no such restaurant.” We do not incorporate belief tracking because we want to fit different dialog tasks in which there may be no annotations of belief states. Also, (Eric and Manning, 2017) pointed out that user utterances already contains the belief state information to generate system response, meaning it is not necessary to explicitly input belief state into the model. For database search, one can train a separate belief tracker with a small portion of data that is annotated, which is more common in the real world cases.

4 Experiments and Results

Data scarcity is one of the biggest challenges in dialog research. It is costly to collect human-human conversations under a specific setting. It is even more time-consuming to annotate the corresponding belief states and dialog acts. With the success of transfer learning in NLP, we aim to mitigate the low-resource problem with the large pre-trained language model. We validate our proposed ARDM on three task-oriented dialog datasets, CamRest676, MulitWOZ, and PersuasionForGood.

4.1 CamRest676

CamRest676 is a relatively small dataset with 408/136/136 dialogs for train/validation/test. We follow Sequicity (Lei et al., 2018) to delexicalize tokens such as restaurant names, phone numbers, postcodes by replacing them with their slot names in utterances. We prepend database search results to the system utterance. An example database search results are “restaurant;3”, where the first slot indicates its dialog domain, which is always “restaurant” in CamRest767, and the second slot represents the number of matched items in the database. We use greedy sampling for all methods during decoding. We use **BLEU-4** and **Success F1** to evaluate language generation quality and Success F1 to evaluate task success. Success F1 computes the F1 score of the generated responses on requested slots such as an address, phone number, or food type. Other than Sequicity, we also compare results by using GPT-2 alone as a language model for the entire dialog. This baseline is equivalent to ARDM without alternating recurrent modeling.

4.1.1 Results

We first test our method on a restaurant search dataset, CamRest676 (Wen et al., 2017). Table 1 shows all models’ results with ground truth belief state or generated belief state. We first use ground truth belief state in all methods to evaluate their response generation quality. ARDM achieves the best BLEU and Success F1 score. We observe that after fine-tuning GPT-2 on the CamRest676, it achieves similar results compared to the previous state-of-the-art method, Sequicity with reinforcement fine-tuning. This suggests pre-trained large-scale language model, such as GPT-2, transfers the meaningful representations to help fine-tuning.

Model	Entity Match rate	Ground Truth Database State BLEU-4	 Success. F1	Generated Database State BLEU-4	 Success. F1
Regular Expression	0.960	-	-	-	-
Sequicity	0.923	21.4	0.852	21.4	0.853
Sequicity (w/o RL)	0.940	22.9	0.821	23.4	0.834
GPT-2-finetune	-	21.8	0.851	19.2	0.862
ARDM	-	**26.2**	**0.864**	**25.4**	**0.862**
ARDM (50% data)	-	25.9	0.859	23.4	0.851

Table 1: Results on CamRest676 dataset. Our method does not require human annotations from dialog states and dialog acts in comparison to Sequicity.

However, without the alternating mechanism, GPT-2 alone does not perform as well as ARDM in terms of both BLEU-4 and Success F1, especially in BLEU-4 (improved 19%). Without the recurrent modeling, the model blends two speakers language distribution and ignores the role difference. Moreover, to test if our model preserves its performance with even less training data, we reduce the training data to 50%, and the performance only drops slightly. With half of the training data, our method still performs significantly better than Sequicity. This result suggests ARDM is robust on low-resource settings due to the advantage of the large-scale pre-training language model.

We also evaluate all models with generated belief states instead of ground truth belief states. Sequicity generates belief tracker results, and its Entity Match rate is 0.927. Our model does not have a state tracker, so we write a separate simple regular expression to extract the occurrence of entities that appear in the database to support our model. Such state tracker achieves 0.960 in Entity Match rate. It suggests that state tracking may be accomplished in more straightforward ways other than training a neural network model on a large set of annotated data. With a simple state tracker, our proposed method still performs better than Sequicity, which trains the belief state and the response generation task jointly.

4.2 MultiWOZ

Here, we only use the ground truth database search result to be consistent with other methods. We perform delexicalization which is mentioned in the original MultiWOZ (Budzianowski et al., 2018). We prepend the database search results to the system response for as conditional input. Also, the database results now contain information about whether the booking is successful or not (i.e., succeed or fail). Again, we prefix the database results to the system response. Note that we do not use belief state or dialog act annotation provided by the dataset to train ARDM. We set the top-p to 0.2 and the temperature to 0.7.

We normalize the time's slot value in all dialogs into the 24-hour format and perform tokenization via spaCy[2]. We found that different papers report results with different versions of the evaluator, which makes it difficult to compare different methods fairly. We explain the differences among all versions of the evaluator in Appendix. In this paper, we follow LaRL's evaluator implementation, as it is more reasonable than others. We re-evaluate results for all methods with the same evaluator to ensure fairness.

We compare our model to the attention-based seq2seq model which is proposed as the MultiWOZ Baseline (Budzianowski et al., 2018), the HDSA (Chen et al., 2019) model that incorporates dialog act supervision as an inductive prior for model architecture, and the LaRL (Zhao et al., 2019) model which leverages latent action modeling and reinforcement learning to improve performance. We do not compare with GPT-2-finetune with our model in MultiWOZ because GPT-2-finetune's performance on CamRest676 is significantly worse than our model. Note that GPT-2-finetune is equivalent to ARDM without alternating parameterization.

The results are evaluated on BLEU-4, Inform Rate, and Success Rate. Inform and Success Rate measure whether the system response provides the recommendations and requested information given in the goal.

[2] https://spacy.io/

Model	Supervision Dialog State	 Dialog Act	Inform (%)	Success (%)	BLEU-4
Human	-	-	98.9	96.5	-
Baseline	✓	×	82.5	72.9	18.9
HDSA	✓	✓	**87.7**	73.4	**23.6**
LaRL	✓	×	82.8	**79.2**	12.8
ARDM	×	×	87.4	72.8	20.6

Table 2: Results on MultiWOZ. Supervision denotes whether a model leverages dialog state or/and dialog act annotations. All models use the ground truth dialog state for database search. ARDM without supervision from annotation can still achieve comparable results.

4.2.1 Results

The evaluation results are shown in Table 2. Without any supervision from dialog states or dialog acts, ARDM significantly outperforms the MultiWOZ Baseline and LaRL on BLEU-4 and Inform rate, and is on par with HDSA. However, HDSA uses dialog act supervision and a large pretrained language model, BERT. Our model requires no annotation and can achieve similar results. This suggests our recurrent modeling and large-scale pre-training methods work similarly as the useful dialog act annotations. All the results show that our method's excellent performance remains consistent in multi-domain dialogs.

We analyze the generated responses and find that if multiple domains have appeared in the conversation history, our model tends to make mistakes in answering the right domain for user requests. This finding suggests that the Maximum Likelihood Estimation (MLE) has limitations in directly optimizing the metric, while reinforcement Learning (RL) can hugely improve the task completion in a dialog system. This is why LaRL has a higher Success rate. However, we also observe that LaRL has a low BLEU-4 score, which indicates low readability in responses. Therefore, there is a trade-off between the generation quality and the task success rate in the RL setting.

4.3 PersuasionForGood

To showcase ARDM's performance on a dialog dataset where it is much more difficult to obtain belief states and dialog act annotations, we train and evaluate our model on PersuasionForGood (Wang et al., 2019) dataset. In this dataset, the persuader must persuade an assigned persuadee (i.e., a person who is asked to donate) to donate money (from their task payment) to a charity called "Save the Children".

4.3.1 Setting

This dataset has a much larger vocabulary size (8,141) than the previous task-oriented dialog datasets due to its non-collaborative dialog property. The conversation content is richer because two speakers are negotiating back and forth. The dataset consists of 1,017 dialogs where only 300 dialogs are annotated with dialog acts. Therefore, models that require dialog state or dialog act annotation are not applicable in this dataset. As ARDM does not require dialog acts for training, it fits well on the PersuasionForGood dataset. We remove the prefix for system inputs because the task does not involve any database search. We use TransferTransfo (Wolf et al., 2019) model as a baseline, since it does not require extra human annotations. TransferTransfo is based on large pre-trained language model, and it uses token type embedding to encode role information of the speaker. We fine-tune it on the PersuasionForGood dataset.

To generate diverse responses, we decode the response using the nucleus sampling (Holtzman et al., 2019) with a top-p of 0.9 and a temperature of 0.7. It is impossible to conduct an automatic evaluation on task success on this task due to the lack of annotation. We use perplexity to evaluate each model's language generation quality. We also conduct a human evaluation to validate each model's task success rate. We show some generated examples in the Appendix to provide more information on both models' generation quality.

4.3.2 Results

Table 3 shows the results for PersuasionForGood. Because ARDM applies better speaker modeling and recurrence mechanism, our model achieves

	Perplexity ↓	Human Preference ↑	Average Donation Amount ↑
TransferTransfo	19.9	34.7%	0.538
ARDM	**10.1**	**65.3%**	**0.807**

Table 3: Automatic Evaluation and Human Evaluation Results on the PersuasionForGood dataset.

lower perplexity compared to TransferTransfo. We did not report BLEU, because on the PersuasionForGood dataset, BLEU score cannot reflect the actual generation quality as a random sentence with common tokens "the, of, is, are..." already has 10.0+ BLEU-1 score. Also because the validation set only contains 100 samples, the result can have a high variance. Then the only way to evaluate this type of system is by human evaluation.

To comprehensively evaluate each model's performance, we recruit 14 human evaluators to chat with the two persuasive systems ten times to avoid the randomness produced by each model. In total, we collected 140 ratings. We ask them to select a preferred chat-bot and indicate how much they are willing to donate after talking to the chat-bot. As a result, human judges prefer ARDM over TransferTransfo and tends to donate more when talking to ARDM produced chat-bot. Our model achieved 27% more donations compared to TransferTransfo. This indicates that our systems are more persuasive. In some examples, such as the one in Table 4, our model generates coherent, natural, and persuasive responses.

5 Error Analysis

Since CamRest676 is similar to MultiWOZ in terms of task content and dialog structure, we only describe the errors in MultiWOZ for simplicity. We randomly selected 30 generated error responses from our model with zero inform and success score.

To our surprise, we observed that nearly 63.3% of errors are not really mistakes. It is mainly due to the limitation of the automatic evaluator. For example, at turn one, the user asks about a restaurant, and the ground truth system response is "the [restaurant_name] is located at ...", but the generated system response is "what food preference do you have?". Our generated response is correct with respect to the dialog context. It is narrowing down the restaurant choices before providing a restaurant recommendation. However, the evaluator sticks to the only possible response it has. Unless the user can dynamically interact with the system, there is no good way to change such mistakes in the automatic evaluator.

We find that another 20% errors our model makes are when the system asks information the user already provided. This type of errors calls for a better history representation. Another 10% errors are due to ignoring the user's request for information, such as phone number. However, when we look at the ground truth responses, some crowd workers also made such errors. So resolving these errors requires a cleaner training dataset. Finally, the rest of 6.7% errors are about incorrect dialog domain understanding. For example, the user is asking for a hotel, but we present a restaurant recommendation. This is because of the data noise during the delexicalization process in which some domain labels are wrong.

The donation persuasion system trained with TransferTransfo and our model has some common problems, such as inconsistency, lack of logic, and hallucination. For example, if the persuader provides the information about "Save the Children", then the persuadee asks "Can you tell me more about it?". The system ends up providing the same information as before. It also sometimes makes up facts that have never happened, such as "Save the Children has an operation about a hurricane in Hawaii". All those errors would prevent users from trusting the bot, and therefore resulting in less donation. However, we also observe that users have a higher tolerance for errors in the persuasion setting than the customer service setting.

Overall, our model performs better on PersuasionForGood by having longer and diversified utterances. This suggests our model which utilizes alternating parameters is more effective than TransferTransfo.

6 Discussions and Ethical Consideration

ARDM models speakers separately on top of a large pre-trained language model. Such simple adaptation demonstrates substantial performance gain. We suspect it is because the interleaved structure of two language models provides a collabora-

Role	Utterance
SYS	Hi, how are you doing today?
USR	I am good!
SYS	Have you ever heard of the charity Save the Children?
USR	No, I have not heard about it. Would you like to talk about it?
SYS	Save the Children is an international non-governmental organization that promotes children's rights, provides relief and helps support children in developing countries.
USR	That's cool. It is an organization for the good of this world.
SYS	Yes, it is. In the first two months of 2018 alone, 1,000 children were reportedly killed or injured in intensifying violence in Syria.
USR	That's sad.
SYS	We do need to do something about that. One of the things that we can do is make sure that we have a system in place that will ensure that every penny we spend goes directly to the children in need.
USR	I agree.
SYS	How much do you like to donate to the charity now? Your donation will be directly deducted from your task payment. You can choose any amount from $0 to all your payment ($2).
USR	I am willing to donate full my payment $2.
SYS	Wow, that is awesome! That is so kind of you!
USR	You are welcome.

Table 4: A conversation between a machine persuader trained by ARDM and a human persuadee.

tive learning frame of both the user and the system language distribution modeling. The memory is the only way for the user and system to communicate, as they do not share any weights in their networks. Thus, the user encoder needs to learn useful representations to make the system model for understanding its intent. Similarly, the system needs to do the same for the user model to improve its understanding. This alternative repeating process forces both the user and system models to preserve the dialog history effectively in the memory. One can interpret the memory as the implicit representation of belief states or dialog acts.

Another benefit of ARDM is that we will obtain both user and system utterance generators. We can let the two models talk to each other to generate new self-play dialogs (Silver et al., 2017). With self-play, one can rapidly build a large scale dialog dataset using adversarial filtering (Zellers et al., 2018). Such models can also potentially be used in reinforcement learning as user simulator to study complex dialog strategies as well. We will explore those possibilities in the future work.

Persuasion is a double-edged sword. Given the fast development of dialog systems, an ethical design principle must be in place throughout all stages of the development and evaluation. We choose the donation task is because it is a relatively simple task that benefits children. Second, when deploying the persuasive agents in real conversations, we need to keep the users informed of the nature of the system. By revealing the identity of the persuasive agent, the user should also have options to communicate directly with the human team behind the system. Lastly, by investigating persuasive dialog systems, we also envision to use them as an educational tool for the general public to learn to defend themselves against machine persuasion.

7 Conclusions

We propose to build Alternating Recurrent Dialog Model (ARDM), a simple, general, and effective dialog method that models user and system separately with large-scale pre-trained language models. Since ARDM does not require any annotations, it generalizes to different dialog applications. Experimental results on CamRest676 and MultiWOZ suggest that ARDM outperforms or is on-par with the current state-of-the-art methods that use manual annotation information, such as belief states and dialog acts. Furthermore, we find our model's excellent performance generalizes to more complex non-collaborative dialog settings. It can generate high-quality responses to persuade people to donate to charity. However, the easiness of training ARDM raises concerns about the misuse of the model in scenarios such as sales, harassment, or scam on a mass scale. We caution the public in deploying such systems in the real world.

References

Pawel Budzianowski and Ivan Vulic. 2019. Hello, it's GPT-2 - how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. *CoRR*, abs/1907.05774.

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 (Riloff et al., 2018), pages 5016–5026.

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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3696–3709. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. 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 2978–2988. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Saarbrücken, Germany, August 15-17, 2017*, pages 37–49. Association for Computational Linguistics.

Mihail Eric and Christopher D. Manning. 2017. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. *CoRR*, abs/1701.04024.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the SIGDIAL 2014 Conference, The 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 18-20 June 2014, Philadelphia, PA, USA*, pages 263–272. The Association for Computer Linguistics.

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

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. Association for Computational Linguistics.

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*. OpenReview.net.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. OpenAI.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors. 2018. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR*, abs/1712.01815.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. *CoRR*, abs/1906.06725.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve J. Young. 2017. Latent intention dialogue models. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3732–3741. PMLR.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *CoRR*, abs/1901.08149.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In (Riloff et al., 2018), pages 93–104.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? 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 2204–2213. Association for Computational Linguistics.

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.