Goal-Oriented End-to-End Conversational Models with Profile Features in a Real-World Setting

Yichao Lu **Manisha Srivastava** **Jared Kramer**
Heba Elfardy **Andrea Kahn** **Song Wang** **Vikas Bhardwaj**
Amazon
Seattle, WA, USA
{yichaolu,mansri,jaredkra,helfardy,kahna,sonwang,vikab}
@amazon.com

Abstract

End-to-end neural models for goal-oriented conversational systems have become an increasingly active area of research, though results in real-world settings are few. We present real-world results for two issue types in the customer service domain. We train models on historical chat transcripts and test on live contacts using a human-in-the-loop research platform. Additionally, we incorporate customer profile features to assess their impact on model performance. We experiment with two approaches for response generation: (1) sequence-to-sequence generation and (2) template ranking. To test our models, a customer service agent handles live contacts and at each turn we present the top four model responses and allow the agent to select (and optionally edit) one of the suggestions or to type their own. We present results for turn acceptance rate, response coverage, and edit rate based on approximately 600 contacts, as well as qualitative analysis on patterns of turn rejection and edit behavior. Top-4 turn acceptance rate across all models ranges from 63%-80%. Our results suggest that these models are promising for an agent-support application.

1 Introduction

While interest in training conversational models has been steadily increasing, recent research has largely focused on how algorithmic improvements help model performance on fabricated datasets. In this work, we explore how two general approaches to conversational modeling perform on real-world data in a customer service setting, with the goal of developing an application that provides customer service agents with recommended responses to enable them to help customers more quickly.

Conversational systems are typically divided into two broad categories: chit-chat systems (Zhang et al., 2018b; Du et al., 2018) and goal-oriented systems, with the latter designed to accomplish specific tasks such as restaurant reservations (Gupta et al., 2018; Peng et al., 2017). Many of these systems use separate components for dialog state (belief) tracking and natural language generation (Bordes et al., 2016; Liu and Lane, 2017; Wen et al., 2016), while others are end-to-end.

Recent work in end-to-end approaches can be further categorized into sequence-to-sequence models that generate responses word by word (Vinyals and Le, 2015; Serban et al., 2016), and template-based approaches that score pairs of conversation history and candidate response (Liu et al., 2018; Zhou et al., 2018; Kannan et al., 2016). Sequence-to-sequence models are easily scaled to new domains but tend to generate safe, generic responses that may not be effective in helping the user achieve their goal (Baheti et al., 2018; Shao et al., 2017; Zhang et al., 2018a; Li et al., 2016). End-to-end templated models are less expensive to develop than belief tracking systems, but still require domain knowledge and may not cover all situations. As a result, some research has focused on hybrid approaches (Qiu et al., 2017; Serban et al., 2017).

Regardless of the approach, there is a shortage of work investigating the performance of task-oriented conversational systems in a real-world setting. The majority of the literature on task-oriented models describes performance on fabricated datasets, which may not translate to real-world data (Gangadharaiah et al., 2018). Furthermore, research on sequence-to-sequence models often borrows automated evaluation metrics from the Machine Translation literature such as BLEU score (Papineni et al., 2002), which have been shown to correlate only weakly with human judgment (Liu et al., 2016; Novikova et al., 2017). In addition, there has been only limited research on using customer profile information to augment textual features in end-to-end conversational models (Zhang et al., 2018b).

In this work, we seek to address these gaps by training task-oriented, multi-turn conversational models for the customer service domain and reporting results on interactions with real customers. Our use case is an agent-support application that recommends responses, similar to the smart reply feature in Gmail (Kannan et al., 2016) and LinkedIn (Pasternack et al., 2017). We incorporate user profile information as features to enable our models to generate relevant, personalized responses. Our research goals are to explore the effect of customer profile features in two different model formulations, and to present a practical comparison of these two general approaches in a real-world setting.

In Section 2, we describe the training data used to develop these models. In Section 3, we describe the end-to-end models we developed, which include both sequence-to-sequence and template-based models. In Section 4, we introduce a research platform for testing our models in a real-world, human-in-the-loop setting, and in Section 5, we report results on conversations with real customers. In Section 6, we describe the impact of profile features, conduct an analysis of rejected and edited turns, and comment on the performance of each type of model. In Section 7, we summarize the conclusions drawn from this work and propose future directions.

2 Data

Our training data consists of historical transcripts from customer service chats handled in English. When a customer begins a chat contact, they are prompted for an initial description of their issue and are then connected to a customer service agent (CSA). We classify these initial utterances into specific customer issues (e.g., *order tracking*, *payment questions*) before connecting the customer to a CSA. These two conversational participants work together to diagnose and resolve the customer's issue(s). Throughout this process, the CSA has access to a wide variety of customer information, such as order status, and internal APIs to execute actions such as canceling or refunding an order. For our experiments, we select the customer issue types *cancel order* and *return refund status*.

Raw text:

Customer: I want to cancel the shoes I ordered yesterday.

Agent: Welcome to Customer Service.

Agent: I am here to help you.

Agent: Give me a moment to look into this.

Training Sample:

Input: **CUSTOMER** I want to cancel the shoes I ordered yesterday. **AGENT** Welcome to Customer Service. **AGENT** I am here to help you. **PROFILE** cancellable, carrier, membership-status. **Output**: Give me a moment to look into this.

Figure 1: Training sample for generative model with profile features. Given raw text and recorded profile features, the training instance is created by appending the turns and profile features while the label is the next agent turn.

In addition to chat transcripts, we experiment with adding particular pieces of customer profile information to our data for the *cancel order* experiments. These data points are similar to the information a CSA accesses when handling a contact. For the *cancel order* related contacts only, we incorporate data points extracted from a customer's profile and details about their orders that are relevant for order cancellation. Specifically, we include the customer's membership status, the order fulfillment method, the shipping carrier, whether the order is a single or multi-item order, and whether the order was eligible for cancellation at the time of contact.

3 Approach

As noted above, our motivation for these experiments is to report real-world results for existing response generation methods and to assess the impact of augmenting our training data with customer profile information. To this end, we train models representing two established response generation strategies: sequence-to-sequence response generation and response ranking. Each model is trained on approximately 5M conversation-response pairs from roughly 350K historical chats related to the specific customer service issue. In this section, we describe our model design and how we incorporate profile information into the system. We train both types of models for the *return refund status* and *cancel order* issue types, but we only add profile information for the latter.

3.1 Response Generation Model

Our response generation models use a transformer-based encoder-decoder model (Vaswani et al., 2017) to create responses. When generating a response at the t-th turn, the encoder takes as input the conversation history $U_t = (u_1, u_2, \dots, u_t)$, where $u_k = (w_1, w_2, \dots, w_{n_k})$ is the utterance k with n_k words.

Additionally, since customer profile information is crucial for correctly diagnosing and solving a customer's issue, we incorporate profile features for the *cancel order* response generation model. We do so by appending the profile features to the conversation history and thus the input to the encoder is of the form $[U_t|P]$, where P represents the profile features and | is the concatenation operator.

Figure 1 shows an example of how the conversation is processed for training. Each turn in the conversation history is prepended by a special token to indicate whether it is an agent or a customer turn, and another token is used to separate profile features from dialogue turns. The embeddings of both words in the turns and customer profile features as in Figure 1 are used as an input to the transformer encoder. The output of the encoder, $S = (s_1, s_2, \dots, s_t)$ where each s_k is a vector of size equal to the dimension of the hidden state of the encoder, is then fed as an input to the decoder together with the target sequence shifted by one word.

3.2 Response Ranking Model

For our response ranking models, the input to the model is a pair consisting of the conversation history with customer profile features and a candidate response. The model outputs a probability of how appropriate a candidate response is given the conversation history. Candidate responses are pulled from a predefined pool of utterances–templates– and ranked based on the probability score. To develop these templates, we use historical chat transcripts between customers and CSAs to come up with a list of approximately 100 templates–50 for each domain–that cover the most common use cases. The advantage of this approach is that it gives us control over how diverse and informative we want our responses to be and allows us to add new templates without having to retrain the model.

This approach uses a hierarchical encoder to encode the context and a separate encoder to encode the response. Each of: (1) last turn, (2) profile features and (3) the rest of the conversation history are encoded using transformer encoders. The outputs of the transformer encoders (X_{last},$X_{profile}$,X_{other}) are then passed through a cross attention layer. A separate word context vector (c_{last}, $c_{profile}$, c_{other}) is passed to each of the cross attention layers. The word context vector can be seen as a high-level representation of a fixed query *"What is the informative word?"* (Yang et al., 2016). Each word context vector is randomly initialized and jointly learned during the training process. The final context encoded vector, given in Equation 1, is the concatenation of the output of the three attention layers passed through a multilayer perceptron.

$$h_{context} = MLP([h_{last}; h_{profile}; h_{other}]) \quad (1)$$

Similar to the context, we use a transformer encoder to encode the candidate response and pass the output through a cross attention layer together with a response context vector c_r. The response context vector is randomly initialized and jointly learned. The classification layer is simply the dot product of the context vector and the response vector followed by a sigmoid activation layer. The output of this layer, given in Equation 2, can be interpreted as how appropriate the candidate response is given the context.

$$score = sigmoid(h_{context} \cdot h_{response}) \quad (2)$$

To create positive training samples, we pair a conversation history with the next agent response. For negative samples, we randomly pick a different conversation history and pair it with the same agent response. Figure 2 shows the architecture of this model.

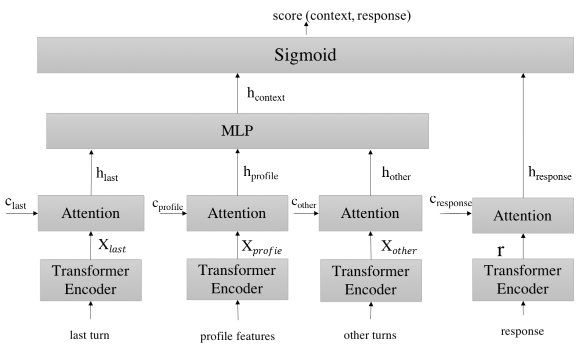

Figure 2: Response Ranking Model Architecture

4 Experimental Setup

We trained response generation and response ranking models using only the conversation history for

```
Repeat until the contact is resolved:
  Every time the customer types into their chat window:
    1. E2E model suggests four responses.
       CSA may wait for customer if still typing.
    2. CSA selects best response & may choose:
       • to edit response
       • "None of the above" & create custom response
    3. CSA sends best response
    4. CSA may choose to repeat steps 1-3 to send
    multiple responses before the next customer utter-
    ance is issued.
```

Figure 3: Contact Flow

return refund status. For *cancel order* we trained both model types with and without profile information, giving us a total of 6 experiments. Our experiments were carried out using a human-in-the-loop research platform that CSAs used to resolve live customer contacts. This platform presents the CSA with the standard chat interface but replaces the CSA's text box with a list of suggested responses from a given model. Every time the customer or the CSA enters text into their chat window, our model refreshes the top four recommended next CSA utterances. Because our goal is to make agents more efficient without degrading the customer experience, we allowed agents to edit suggested turns before sending them. Figure 3 shows the conversational flow for each chat contact.

5 Results

We describe the results of our experiments in terms of turn acceptance, defined as the percentage of agent turns for which a model-generated response was accepted by the CSA. We present results for *cancel order* and *return refund status* separately because the issues do not have the same resolution difficulty. Qualitative analysis on reasons for turn rejection and patterns of edit behavior are included in Section 6.

In Table 1, we present results for 613 contacts handled by our models with a human in the loop (approximately 100 contacts per experiment). In each of our experiments, the CSA was shown the top 4 model suggestions at each turn; turn acceptance at 4 refers to the frequency with which the CSA chose one of the suggestions, and turn acceptance at 1 refers to the frequency with which the CSA chose the highest ranked suggestion. The average number of CSA turns is roughly equal for all experiments.

Turn acceptance at 4 ranges from 63.0% to 80.1%, with the generative+profile model achieving the best performance for *cancel order* and the generative model performing best for *return refund status*. Turn acceptance at 1 ranges from 27.1% to 38.8%, with the ranking+profile model performing best for *cancel order* and the ranking model performing best for *return refund status*. It is possible that turn acceptance at 1 would be higher in a setting where agents were not given 4 choices.

We also look at our models' ability to handle contacts start to finish and report the percentage of contacts for which all or all but one of the agent turns were model-generated. For these metrics, the top performing models are generative+profile for *cancel order* (14.9%) and generative for *return refund status* (26.3%). For both issue types, only a very small number of contacts ($<$10) had the top recommendation selected for all turns. The average depth of the first rejection (the percentage of turns in the conversation that occurred before the first rejection) was also highest for the generative+profile model for *cancel order* (68%) and the generative model for *return refund status* (48%). We also found that, on average, accepted turns are 4-5 words shorter than rejected ones, suggesting that longer, more complex turns are less likely to be accepted.

6 Analysis & Discussion

In this section, we describe qualitative analysis on observed patterns in edit behavior and turn rejection for each experiment. We also compare the performance of the generative vs. the ranking models and describe the impact of adding profile features to the models.

6.1 Edited Turns

Since we allowed CSAs to edit suggested turns before sending them, we calculated the edit rate and analyzed the nature of edits they performed. Our goals in this analysis were to 1) estimate the percentage of accepted turns that require only minor, stylistic changes, vs. those that require more substantial edits and 2) understand the nature of major edits so we can improve model recommendations in these cases.

	Cancel Order				Check R/R Status	
	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.6
	Gen	Gen+profile	Rank	Rank+profile	Gen	Rank
Total contacts	123	74	124	79	114	99
Average number of CSA turns	12.7	9.9	11.2	10.2	11.4	11.4
%Contacts: all turns accepted	8.5	**14.9**	8.9	12.7	**26.3**	11.1
%Contacts: all but 1 turn accepted	21.1	**39.2**	21.8	29.1	**21.9**	16.2
%Contacts: all turns accepted with top recommendation	0.0	**5.4**	0.0	1.3	0.9	**2.0**
Turn acceptance at 4	74.7	**80.1**	63.0	76.3	**75.5**	70.7
Turn acceptance at 1	33.3	37.2	33.7	**38.8**	27.1	**33.2**
Average depth of first rejection (% of contact)	53	**68**	55	60	**48**	47
Median depth of first rejection (% of contact)	50	**67**	50	60	**46**	45
Avg. no. of words in rejected turns	15.5	12.0	14.5	11.0	10.3	13.4
Avg. no. of words in accepted turns	8.5	7.7	7.6	7.3	9.3	7.8

Table 1: Results of the different experiments on both *cancel order* and *return refund status* domains.

We calculated token-level edit distance[1] between suggested and sent turns (see Table 2). We then grouped the accepted turns into three buckets with edit distances of 0, 1-5, and >5, respectively, based on observations that edits of length <5 were more often stylistic and edits of length >5 more often changed the fundamental message.

We further analyzed a random sample of 100 turns with edits of length >5 and found that they fell into four broad categories:

1. The CSA replaced a conversation filler with a more informative response. The suggestion that was edited would not have derailed the conversation, but it also did not directly progress toward the customer's goal. This was particularly common for the generative models. We observe a similar trend in the rejected turns (see Section 6.2).
2. The recommended text provided extra information (which may have been incorrect or redundant) that the CSA deleted. This was observed only for the ranking models and suggests that these models would benefit from shorter templates.
3. The recommended text provided some information but not all, and the CSA needed to edit it to add something. This was observed only for the generative models.
4. The CSA replaced a conversational filler with a different conversational filler. This was observed across both types of models.

One surprising observation was that the number of accepted turns with no edits was greater for the models that did not incorporate profile information. This requires deeper analysis.

6.2 Rejected Turns

We performed a manual analysis of 200 rejected turns across all experiments to understand how we can improve the performance of our models. We found that turn rejections were occasionally caused by real-world constraints that our models were not prepared to handle. For example, in 7% of the cases the CSA needed to reject the model's suggestions in order to send an idle message asking if the customer was still present. In future work, this could be addressed by incorporating idle time as a contextual feature.

Both generative and ranking models were able to handle greetings and closings fairly well. However, the models struggled at times with the crucial turns in our goal-oriented setting: those where the bot must suggest an actual resolution to the customer's issue. As noted above, the average depth of rejection was roughly halfway through the contact. Given that the first quarter of a contact is primarily greetings and the second quarter is typically diagnosis, our models seem to fail most often during the solution phase of the contact. In our analysis, we found that issue breadth is a common cause of turn rejection, with 24% of turn rejection coming from cases such as canceling subscriptions or gift cards instead of physical orders.

[1] We use a token-level version of Levenshtein edit distance between the two case-normalized strings after stripping punctuation. Additions, deletions, and substitutions are all counted equally.

	Cancel Order				Check R/R Status	
	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.6
	Gen	Gen+profile	Rank	Rank+profile	Gen	Rank
Accept. with no edits (%)	45.5	39.0	42.4	34.5	34.3	42.2
Accept. with edit distance >1 and <5 (%)	26.3	29.4	25.5	22.1	20.7	17.3
Accept. with edit distance >5 (%)	28.2	31.6	32.1	43.5	45.0	40.5

Table 2: Rate of edit distances between accepted and sent turns

An additional 4% of turn rejections occurred when the contact needed to be transferred to a specialist. Providing additional order-specific profile features could help the model navigate these edge cases.

Agent behavior and training was also a major factor in turn acceptance. We found that 16% of rejected turns occurred when the CSA decided to reject the model suggestions in favor of a stylistically different but semantically similar message. We also found that 34% of the turn rejections could have potentially been avoided if the agent had been willing to take sequential turns. For example, the models frequently suggest turns such as *Thanks, just a moment* or *I see*, which the agent rejected in favor of a more goal-oriented statement. However, accepting one of these suggestions would not have interrupted the conversation flow and would have given the model another chance to produce the appropriate suggestion.

6.3 Generative vs. Ranking Models

Across all experiments, the generative models outperformed the ranking models in terms of top-4 turn acceptance rate. The best performing models in terms of per-contact response coverage (all turns accepted and all but 1 turn accepted) and depth of first rejection were also generative models. The ranking models, however, performed best in terms of top-1 turn acceptance. A closer analysis of rejected turns from the generative and ranking models showed that the relatively higher acceptance rate of the generative models came from both greetings/closings and resolution-specific turns. Suggestions from the generative models did lack semantic diversity compared to the ranking models. This is because the generative models perform beam search at the word level as opposed to utterance level for the ranking models. We expected that this would give the ranking models an advantage in turn acceptance rate, but this was not borne out. This may be due to the fact that CSAs seem to reject messages that are too specific or too long; rejecting the entire message and writing a new one is more appealing than editing for longer model suggestions.

6.4 Profile Features

Adding profile features improved both top-1 and top-4 turn acceptance rate for both generative and ranking models. As one would expect, we observed that these features altered the models' suggested responses. When the membership-status feature was positive, the models consistently suggested appropriate greetings such as *Thanks for being a member, how can I help you.* When the cancel-eligible feature was positive, the model was more likely to suggest *Sure. I can cancel the order for you*, and when this feature was negative the model made suggestions such as *I am sorry I cannot cancel your order since your order has shipped and entered the shipping process*.

6.5 User Feedback

Since our goal is to help agents resolve customers' issues more efficiently, we asked agents to provide us with their feedback on the system. The feedback was overwhelmingly positive, with almost all agents reporting that having personalized response recommendations saved them time and that customers were also impressed by how quickly the agents were able to fix their issues. Some agents also reported that they liked that they did not have to type, which suggests that the proposed system could be used as an accessibility tool.

7 Conclusion & Future Work

In this paper, we present experimental results for generative and ranking end-to-end models for real-world goal-oriented conversations. We trained our models using both transcript and customer profile data. Our models achieve a top-4 turn acceptance rate ranging from 63% to 80%, suggesting that these models can be effective in assisting CSAs by recommending text responses as they handle customer chats. Agent feedback on the recommendations from these models has been overwhelmingly positive, and such an agent-support application has the potential to improve customers' experience by enabling agents to assist them more efficiently.

We believe that additional investment in contextual and profile features would improve performance for both model types. In addition, we could improve the performance of the ranking models by modifying their template pools, primarily by adding shorter utterances or splitting longer utterances. In future work, combining output from the generative and the ranking models in a single system could improve overall performance.

Lastly, this work highlights the need for a more mature rubric for analysis of turn rejection and edit reasons. We observed CSAs rejecting and editing turns to add and remove information, and to avoid particular conversational fillers. We also observed model failures due to real-world constraints like customers going idle and issue-specific edge cases. In future work, we plan to develop a more comprehensive annotation guide for error analysis in real-world goal-oriented systems.

References

Ashutosh Baheti, Alan Ritter, Jiwei Li, and Bill Dolan. 2018. Generating more interesting responses in neural conversation models with distributional constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. In *arXiv preprint arXiv:1605.07683*.

Jiachen Du, Wenjie Li, Yulan He, Ruifeng Xu, Lidong Bing, and Xuan Wang. 2018. Variational autoregressive decoder for neural response generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Rashmi Gangadharaiah, Balakrishnan Narayanaswamy, and Charles Elkan. 2018. What we need to learn if we want to do and not just talk. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 25–32. Association for Computational Linguistics.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, Laszlo Lukacs, Marina Ganea, Peter Young, et al. 2016. Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 955–964. ACM.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Bing Liu and Ian Lane. 2017. An end-to-end trainable neural network model with belief tracking for task-oriented dialog. In *Interspeech*.

Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2060–2069.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Jeff Pasternack, Nimesh Chakravarthi, Adam Leon, Nandeesh Rajashekar, Birjodh Tiwana, and Bing Zhao. 2017. Building smart replies for member messages.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. 2017. Alime chat: A sequence to sequence and rerank based chatbot engine. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 498–503.

Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, et al. 2017. A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, pages 3776–3784.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

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.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. In *arXiv preprint arXiv:1604.04562*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2018a. Learning to control the specificity in neural response generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018b. 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 (Volume 1: Long Papers)*.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.