Do Neural Dialog Systems Use the Conversation History Effectively? An Empirical Study

Chinnadhurai Sankar[1,2,4*] **Sandeep Subramanian**[1,2,5]

Christopher Pal[1,3,5] **Sarath Chandar**[1,2,4] **Yoshua Bengio** [1,2]

[1]Mila [2]Université de Montréal [3]École Polytechnique de Montréal
[4]Google Research, Brain Team [5]Element AI, Montréal

Abstract

Neural generative models have been become increasingly popular when building conversational agents. They offer flexibility, can be easily adapted to new domains, and require minimal domain engineering. A common criticism of these systems is that they seldom understand or use the available dialog history effectively. In this paper, we take an empirical approach to understanding how these models use the available dialog history by studying the sensitivity of the models to artificially introduced *unnatural* changes or perturbations to their context at test time. We experiment with 10 different types of perturbations on 4 multi-turn dialog datasets and find that commonly used neural dialog architectures like recurrent and transformer-based seq2seq models are rarely sensitive to most perturbations such as missing or reordering utterances, shuffling words, etc. Also, by open-sourcing our code, we believe that it will serve as a useful diagnostic tool for evaluating dialog systems in the future [1].

1 Introduction

With recent advancements in generative models of text (Wu et al., 2016; Vaswani et al., 2017; Radford et al., 2018), neural approaches to building chit-chat and goal-oriented conversational agents (Sordoni et al., 2015; Vinyals and Le, 2015; Serban et al., 2016; Bordes and Weston, 2016; Serban et al., 2017b) has gained popularity with the hope that advancements in tasks like machine translation (Bahdanau et al., 2015), abstractive summarization (See et al., 2017) should translate to dialog systems as well. While these models have demonstrated the ability to generate fluent responses, they still lack the ability to "understand" and process the dialog history to produce coherent and interesting responses. They often produce boring and repetitive responses like "Thank you." (Li et al., 2015; Serban et al., 2017a) or meander away from the topic of conversation. This has been often attributed to the manner and extent to which these models use the dialog history when generating responses. However, there has been little empirical investigation to validate these speculations.

In this work, we take a step in that direction and confirm some of these speculations, showing that models do not make use of a lot of the information available to it, by subjecting the dialog history to a variety of synthetic perturbations. We then empirically observe how recurrent (Sutskever et al., 2014) and transformer-based (Vaswani et al., 2017) sequence-to-sequence (seq2seq) models respond to these changes. The central premise of this work is that *models make minimal use of certain types of information if they are insensitive to perturbations that destroy them*. Worryingly, we find that 1) both recurrent and transformer-based seq2seq models are insensitive to most kinds of perturbations considered in this work 2) both are particularly insensitive even to extreme perturbations such as randomly shuffling or reversing words within every utterance in the conversation history (see Table 1) and 3) recurrent models are more sensitive to the ordering of utterances within the dialog history, suggesting that they could be modeling conversation dynamics better than transformers.

2 Related Work

Since this work aims at investigating and gaining an understanding of the kinds of information a generative neural response model learns to use, the most relevant pieces of work are where sim-

*Corresponding author: chinnadhurai@gmail.com

[1]Code is available at https://github.com/chinnadhurai/ParlAI/

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 32–37
Florence, Italy, July 28 - August 2, 2019.

	No Perturbations	Token shuffling
1	Good afternoon ! Can I help you ?	I afternoon help you Good ? ! Can
2	Could you show me where the Chinesc-style clothing is located ? I want to buy a silk coat	the located Chinesc-style where is show a . buy you ? I clothing want coat silk me Could to
3	This way , please . Here they are . They're all handmade .	are handmade . way please This all Here they . , They're .
4	**Model Response**: How much is it ?	**Model Response**: How much is it ?

Table 1: An example of an LSTM seq2seq model with attention's insensitivity to shuffling of words in the dialog history on the DailyDialog dataset.

ilar analyses have been carried out to understand the behavior of neural models in other settings. An investigation into how LSTM based *unconditional* language models use available context was carried out by Khandelwal et al. (2018). They empirically demonstrate that models are sensitive to perturbations only in the nearby context and typically use only about 150 words of context. On the other hand, in conditional language modeling tasks like machine translation, models are adversely affected by both synthetic and natural noise introduced anywhere in the input (Belinkov and Bisk, 2017). Understanding what information is learned or contained in the representations of neural networks has also been studied by "probing" them with linear or deep models (Adi et al., 2016; Subramanian et al., 2018; Conneau et al., 2018).

Several works have recently pointed out the presence of annotation artifacts in common text and multi-modal benchmarks. For example, Gururangan et al. (2018) demonstrate that hypothesis-only baselines for natural language inference obtain results *significantly* better than random guessing. Kaushik and Lipton (2018) report that reading comprehension systems can often ignore the entire question or use only the last sentence of a document to answer questions. Anand et al. (2018) show that an agent that does not navigate or even see the world around it can answer questions about it as well as one that does. These pieces of work suggest that while neural methods have the potential to learn the task specified, its design could lead them to do so in a manner that doesn't use all of the available information within the task.

Recent work has also investigated the inductive biases that different sequence models learn. For example, Tran et al. (2018) find that recurrent models are better at modeling hierarchical structure while Tang et al. (2018) find that feedforward architectures like the transformer and convolutional models are not better than RNNs at modeling long-distance agreement. Transformers however excel at word-sense disambiguation. We analyze whether the choice of architecture and the use of an attention mechanism affect the way in which dialog systems use information available to them.

3 Experimental Setup

Following the recent line of work on generative dialog systems, we treat the problem of generating an appropriate response given a conversation history as a conditional language modeling problem. Specifically we want to learn a conditional probability distribution $P_\theta(y|x)$ where y is a reasonable response given the conversation history x. The conversation history is typically represented as a sequence of utterances $\mathbf{x_1}, \mathbf{x_2}, \ldots \mathbf{x_n}$, where each utterance $\mathbf{x_i}$ itself is comprised of a sequence of words $x_{i_1}, x_{i_2} \ldots x_{i_k}$. The response y is a single utterance also comprised of a sequence of words $y_1, y_2 \ldots y_m$. The overall conditional probability is factorized autoregressively as

$$P_\theta(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} P_\theta(y_i|y_{<i}, \mathbf{x_1} \ldots \mathbf{x_n})$$

P_θ, in this work, is parameterized by a recurrent or transformer-based seq2seq model. The *crux of this work is to study how the learned probability distribution behaves as we artificially perturb the conversation history* $\mathbf{x_1}, \ldots \mathbf{x_n}$. We measure behavior by looking at how much the per-token perplexity increases under these changes. For example, one could think of shuffling the order in which $\mathbf{x_1} \ldots \mathbf{x_n}$ is presented to the model and observe how much the perplexity of $\mathbf{y}$ under the model increases. If the increase is only minimal, we can conclude that the ordering of $\mathbf{x_1} \ldots \mathbf{x_n}$ isn't informative to the model. For a complete list of perturbations considered in this work, please refer to Section 3.2. All models are trained without any perturbations and sensitivity is studied *only at test time*.

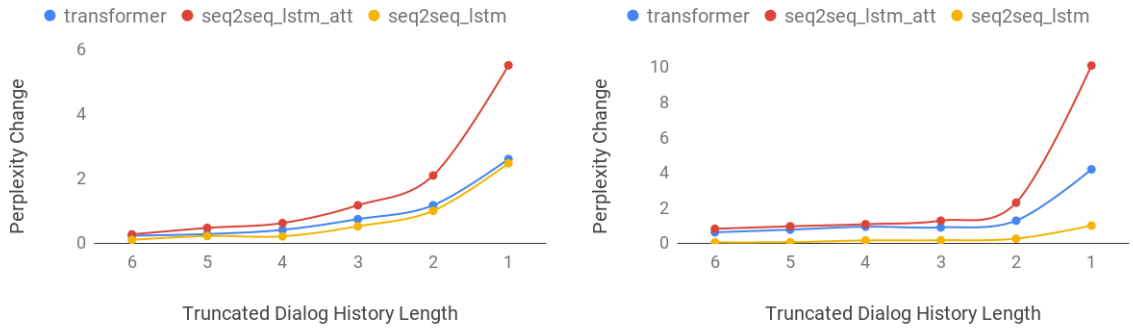

Figure 1: The increase in perplexity for different models when only presented with the k most recent utterances from the dialog history for Dailydialog (left) and bAbI dialog (right) datasets. Recurrent models with attention fare better than transformers, since they use more of the conversation history.

3.1 Datasets

We experiment with four multi-turn dialog datasets.

bAbI dialog is a synthetic goal-oriented multi-turn dataset (Bordes and Weston, 2016) consisting of 5 different tasks for restaurant booking with increasing levels of complexity. We consider Task 5 in our experiments since it is the hardest and is a union of all four tasks. It contains $1k$ dialogs with an average of 13 user utterances per dialog.

Persona Chat is an open domain dataset (Zhang et al., 2018) with multi-turn chit-chat conversations between turkers who are each assigned a "persona" at random. It comprises of $10.9k$ dialogs with an average of 14.8 turns per dialog.

Dailydialog is an open domain dataset (Li et al., 2017) which consists of dialogs that resemble day-to-day conversations across multiple topics. It comprises of $13k$ dialogs with an average of 7.9 turns per dialog.

MutualFriends is a multi-turn goal-oriented dataset (He et al., 2017) where two agents must discover which friend of theirs is mutual based on the friends' attributes. It contains $11k$ dialogs with an average of 11.41 utterances per dialog.

3.2 Types of Perturbations

We experimented with several types of perturbation operations at the utterance and word (token) levels. All perturbations are applied in isolation.

Utterance-level perturbations We consider the following operations 1) *Shuf* that shuffles the sequence of utterances in the dialog history, 2) *Rev* that reverses the order of utterances in the history (but maintains word order within each utterance) 3) *Drop* that completely drops certain utterances and 4) *Truncate* that truncates the dialog history to contain only the k most recent utterances where $k \leq n$, where n is the length of dialog history.

Word-level perturbations We consider similar operations but at the word level within **every** utterance 1) *word-shuffle* that randomly shuffles the words within an utterance 2) *reverse* that reverses the ordering of words, 3) *word-drop* that drops 30% of the words uniformly 4) *noun-drop* that drops *all* nouns, 5) *verb-drop* that drops *all* verbs.

3.3 Models

We experimented with two different classes of models - recurrent and transformer-based sequence-to-sequence generative models. All data loading, model implementations and evaluations were done using the ParlAI framework. We used the default hyper-parameters for all the models as specified in ParlAI.

Recurrent Models We trained a seq2seq (*seq2seq_lstm*) model where the encoder and decoder are parameterized as LSTMs (Hochreiter and Schmidhuber, 1997). We also experiment with using decoders that use an attention mechanism (*seq2seq_lstm_att*) (Bahdanau et al., 2015). The encoder and decoder LSTMs have 2 layers with 128 dimensional hidden states with a dropout rate of 0.1.

Transformer Our transformer (Vaswani et al., 2017) model uses 300 dimensional embeddings and hidden states, 2 layers and 2 attention heads with no dropout. This model is significantly smaller than the ones typically used in machine

Models	Test PPL	Only Last	Shuf	Rev	Drop First	Drop Last	Word Drop	Verb Drop	Noun Drop	Word Shuf	Word Rev
		Utterance level perturbations ($\Delta PPL_{[\sigma]}$)					Word level perturbations ($\Delta PPL_{[\sigma]}$)				
DailyDialog											
seq2seq_lstm	$32.90_{[1.40]}$	$1.70_{[0.41]}$	$\mathbf{3.35}_{[0.38]}$	$\mathbf{4.04}_{[0.28]}$	$0.13_{[0.04]}$	$\mathbf{5.08}_{[0.79]}$	$1.58_{[0.15]}$	$0.87_{[0.08]}$	$1.06_{[0.28]}$	$\mathbf{3.37}_{[0.33]}$	$3.10_{[0.45]}$
seq2seq_lstm_att	$29.65_{[1.10]}$	$\mathbf{4.76}_{[0.39]}$	$2.54_{[0.24]}$	$3.31_{[0.49]}$	$\mathbf{0.32}_{[0.03]}$	$4.84_{[0.42]}$	$\mathbf{2.03}_{[0.25]}$	$\mathbf{1.37}_{[0.29]}$	$2.22_{[0.22]}$	$2.82_{[0.31]}$	$\mathbf{3.29}_{[0.25]}$
transformer	$\mathbf{28.73}_{[1.30]}$	$3.28_{[1.37]}$	$0.82_{[0.40]}$	$1.25_{[0.62]}$	$0.27_{[0.19]}$	$2.43_{[0.83]}$	$1.20_{[0.69]}$	$0.63_{[0.17]}$	$\mathbf{2.60}_{[0.98]}$	$0.15_{[0.08]}$	$0.26_{[0.18]}$
Persona Chat											
seq2seq_lstm	$43.24_{[0.99]}$	$3.27_{[0.13]}$	$6.29_{[0.48]}$	$\mathbf{13.11}_{[1.22]}$	$0.47_{[0.21]}$	$\mathbf{6.10}_{[0.46]}$	$1.81_{[0.25]}$	$0.68_{[0.19]}$	$0.75_{[0.15]}$	$1.29_{[0.17]}$	$1.95_{[0.20]}$
seq2seq_lstm_att	$42.90_{[1.76]}$	$\mathbf{4.44}_{[0.81]}$	$\mathbf{6.70}_{[0.67]}$	$11.61_{[0.75]}$	$\mathbf{2.99}_{[2.24]}$	$5.58_{[0.45]}$	$\mathbf{2.47}_{[0.67]}$	$\mathbf{1.11}_{[0.27]}$	$\mathbf{1.20}_{[0.23]}$	$\mathbf{2.03}_{[0.46]}$	$\mathbf{2.39}_{[0.31]}$
transformer	$\mathbf{40.78}_{[0.31]}$	$1.90_{[0.08]}$	$1.22_{[0.22]}$	$1.41_{[0.54]}$	$-0.1_{[0.07]}$	$1.59_{[0.39]}$	$0.54_{[0.08]}$	$0.40_{[0.00]}$	$0.32_{[0.18]}$	$0.01_{[0.01]}$	$0.00_{[0.06]}$
MutualFriends											
seq2seq_lstm	$14.17_{[0.29]}$	$1.44_{[0.86]}$	$\mathbf{1.42}_{[0.25]}$	$1.24_{[0.34]}$	$0.00_{[0.00]}$	$0.76_{[0.10]}$	$0.28_{[0.11]}$	$0.00_{[0.03]}$	$0.61_{[0.39]}$	$0.31_{[0.25]}$	$0.56_{[0.39]}$
seq2seq_lstm_att	$\mathbf{10.60}_{[0.21]}$	$\mathbf{32.13}_{[4.08]}$	$1.24_{[0.19]}$	$1.06_{[0.24]}$	$0.08_{[0.03]}$	$\mathbf{1.35}_{[0.15]}$	$\mathbf{1.56}_{[0.20]}$	$0.15_{[0.07]}$	$\mathbf{3.28}_{[0.38]}$	$\mathbf{2.35}_{[0.22]}$	$\mathbf{4.59}_{[0.46]}$
transformer	$10.63_{[0.03]}$	$20.11_{[0.67]}$	$1.06_{[0.16]}$	$\mathbf{1.62}_{[0.44]}$	$\mathbf{0.12}_{[0.03]}$	$0.81_{[0.09]}$	$0.75_{[0.05]}$	$\mathbf{0.16}_{[0.02]}$	$1.50_{[0.12]}$	$0.07_{[0.01]}$	$0.13_{[0.04]}$
bAbi dailog: Task5											
seq2seq_lstm	$1.28_{[0.02]}$	$1.31_{[0.50]}$	$\mathbf{43.61}_{[15.9]}$	$\mathbf{40.99}_{[9.38]}$	$0.00_{[0.00]}$	$4.28_{[1.90]}$	$0.38_{[0.11]}$	$0.01_{[0.00]}$	$0.10_{[0.06]}$	$0.09_{[0.02]}$	$0.42_{[0.38]}$
seq2seq_lstm_att	$\mathbf{1.06}_{[0.02]}$	$\mathbf{9.14}_{[1.28]}$	$41.21_{[8.03]}$	$34.32_{[10.7]}$	$0.00_{[0.00]}$	$\mathbf{6.75}_{[1.86]}$	$\mathbf{0.64}_{[0.07]}$	$0.03_{[0.03]}$	$0.22_{[0.04]}$	$\mathbf{0.25}_{[0.01]}$	$\mathbf{1.10}_{[0.80]}$
transformer	$1.07_{[0.00]}$	$4.06_{[0.33]}$	$0.38_{[0.02]}$	$0.62_{[0.02]}$	$0.00_{[0.00]}$	$0.21_{[0.02]}$	$0.36_{[0.02]}$	$\mathbf{0.25}_{[0.06]}$	$\mathbf{0.37}_{[0.06]}$	$0.00_{[0.00]}$	$0.00_{[0.00]}$

Table 2: Model performance across multiple datasets and sensitivity to different perturbations. Columns 1 & 2 report the test set perplexity (without perturbations) of different models. Columns 3-12 report the **increase** in perplexity when models are subjected to different perturbations. The mean (μ) and standard deviation $[\sigma]$ across 5 runs are reported. The *Only Last* column presents models with **only** the last utterance from the dialog history. The model that exhibits the highest sensitivity (higher the better) to a particular perturbation on a dataset is in bold. *seq2seq_lstm_att* are the most sensitive models **24/40** times, while transformers are the least with **6/40** times.

translation since we found that the model that resembled Vaswani et al. (2017) significantly overfit on all our datasets.

While the models considered in this work might not be state-of-the-art on the datasets considered, we believe these models are still competitive and used commonly enough at least as baselines, that the community will benefit by understanding their behavior. In this paper, we use early stopping with a patience of 10 on the validation set to save our best model. All models achieve close to the perplexity numbers reported for generative seq2seq models in their respective papers.

4 Results & Discussion

Our results are presented in Table 2 and Figure 1. Table 2 reports the perplexities of different models on test set in the second column, followed by the **increase** in perplexity when the dialog history is perturbed using the method specified in the column header. Rows correspond to models trained on different datasets. Figure 1 presents the change in perplexity for models when presented only with the k most recent utterances from the dialog history.

We make the following observations:

1. Models tend to show only tiny changes in perplexity in most cases, even under extreme changes to the dialog history, suggesting that they use far from all the information that is available to them.

2. Transformers are insensitive to word-reordering, indicating that they could be learning bag-of-words like representations.

3. The use of an attention mechanism in *seq2seq_lstm_att* and transformers makes these models use more information from earlier parts of the conversation than vanilla seq2seq models as seen from increases in perplexity when using only the last utterance.

4. While transformers converge faster and to lower test perplexities, they don't seem to capture the conversational dynamics across utterances in the dialog history and are less sensitive to perturbations that scramble this structure than recurrent models.

5 Conclusion

This work studies the behaviour of generative neural dialog systems in the presence of synthetically introduced perturbations to the dialog history, that it conditions on. We find that both recurrent and transformer-based seq2seq models are not significantly affected even by drastic and unnatural modifications to the dialog history. We also find subtle differences between the way in which recurrent and transformer-based models use available context. By open-sourcing our code, we believe this paradigm of studying model behavior by introducing perturbations that destroys different kinds of structure present within the dialog history can

be a useful diagnostic tool. We also foresee this paradigm being useful when building new dialog datasets to understand the kinds of information models use to solve them.

Acknowledgements

We would like to acknowledge NVIDIA for donating GPUs and a DGX-1 computer used in this work. We would also like to thank the anonymous reviewers for their constructive feedback. Our code is available at https://github.com/chinnadhurai/ParlAI/.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Ankesh Anand, Eugene Belilovsky, Kyle Kastner, Hugo Larochelle, and Aaron Courville. 2018. Blindfold baselines for embodied qa. *arXiv preprint arXiv:1811.05013*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *In Proceedings Of The International Conference on Representation Learning (ICLR 2015)*.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Antoine Bordes and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *CoRR*, abs/1605.07683.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

H. He, A. Balakrishnan, M. Eric, and P. Liang. 2017. Learning Symmetric Collaborative Dialogue Agents with Dynamic Knowledge Graph Embeddings. *arXiv e-prints*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. *arXiv preprint arXiv:1805.04623*.

J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. 2015. A Diversity-Promoting Objective Function for Neural Conversation Models. *ArXiv e-prints*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, and Y. Bengio. 2017a. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference (AAAI)*.

Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, et al. 2017b. 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 *Proceedings of AAAI*.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why self-attention? a targeted evaluation of neural machine translation architectures. *arXiv preprint arXiv:1808.08946*.

Ke Tran, Arianna Bisazza, and Christof Monz. 2018. The importance of being recurrent for modeling hierarchical structure. *arXiv preprint arXiv:1803.03585*.

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*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

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? *arXiv preprint arXiv:1801.07243*.