Why do you say that?
Rationale extraction for dialogue modelling

Spyros Avlonitis
spyrosavl@gmail.com
University of Amsterdam

Xinyi Chen
xinyi.chen@student.uva.nl
University of Amsterdam

Gerson Foks
gerson.foks@gmail.com
University of Amsterdam

Frank Verhoef
fsa.verhoef@chello.nl
University of Amsterdam

Abstract

The increasing use of neural models in natural language processing (NLP) has created a desire for more interpretability. We investigate the use of a rationale extractor to provide insight into which parts of a dialogue history provide a justification for the predictions of a language model for dialogue generation. Although various methods have been developed for a range of NLP tasks such as sentiment classification or natural language inference, there is no such method yet for dialogue generation. Our experiments show that, of the three variants of rationale extractors we investigate, the best results are obtained when the rationale is extracted using the Gumbel-Softmax trick on the embeddings of the dialogue history. Using this Gumbel based rationale extractor results in responses with similar perplexity as the base model and in some aspects the rationales are selected seems plausible to us. Although our initial results are limited, our experiments provide a good basis for further research by providing an overview of how rationale extractors can be combined with a dialogue generation model.

1 Introduction

Understanding neural models for dialogue generation is still an unresolved challenge. With the growing use of neural models, the need for explainability has also increased. For NLP tasks, the creation of an explanation comes down to the identification of the smallest subset of the input that is still sufficient to generate the desired output.

Several methods have been proposed for this purpose, such as the LIME model (Ribeiro et al., 2016), the attention mechanism (Li et al., 2016) or rationale extraction (Lei et al., 2016). The goal of these models is to indicate which words in the input are most important to generate the model's prediction. These methods have mostly been applied on NLP tasks where the success of a task is well defined, such as classification, natural language inference and translation.

Explainability for dialogue generation poses additional challenges because regular metrics for NLP tasks as accuracy, perplexity, or BLUE scores do not correlate with human judgement on dialogue quality (Liu et al., 2016a). This makes it difficult to judge whether the system that provides the explanation does not harm the quality of the generated responses.

Inspired by the work of Lei et al. (2016) and Bastings et al. (2019), we investigate the combination of a rationale extractor with a dialogue generation model. For the rationale extractor, we look at two types: *i)* one policy-based rationale extractor that optimizes a policy (masking or selecting a word) using the REINFORCE algorithm (Williams, 1992) and uses the language model as a black box; *ii)* a model that integrates the rationale extractor and the language model into one model that share their embedding layer. The rationale is either chosen with the help of the kumaraswamy distribution (Kumaraswamy, 1980) or the Gumbel-softmax trick (Jang et al., 2017). We compare results of these rationale extractors in combination with three language models: a simple LSTM based language model, DialoGPT and RoBERTa.

Our main contributions are as follows [1]:

- To our knowledge rationale extraction has not been used before in combination with a language model. We show how two types of rationale extractors can be combined with a neural models for dialogue generation;
- We analyse the results both quantitatively and qualitatively. Furthermore we discuss the cur-

[1] Code available at https://github.com/gersonfoks/rational-dialog-model

rent shortcommings of the models and propose a path forward.

The barriers we encountered to train the models and evaluate the results underline the fact that dialogue generation is harder than other NLP tasks such as text classification and translation. This indicates the need further advances in training methods and evaluation measures for dialogue models.

2 Related work

The desire for explainability has sparked numerous research initiatives and publications. The LIME model (Ribeiro et al., 2016) explains the predictions of any classifier in an interpretable and faithful manner. LIME treats the classifier as a black box and can therefore also be applied to any NLP classification task.

The attention mechanism that has enjoyed so much popularity since the publication of the paper "Attention is all you need" (Vaswani et al., 2017) has not only provided improved performance in all types of NLP tasks, but has also been used to explain the models' predictions. The attention levels that are associated with each input token are treated as a measure of the importance of that token for the prediction. The use of the attention mechanism for explanation is under debate however (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019).

Lei et al. (2016) pioneered the approach to extract a rationale for the prediction of a text classifier. They assign a binary Bernouilli variable to each input token, compose a rationale of all tokens where a 1 was sampled and optimize the rationale using the REINFORCE algorithm (Williams, 1992). Their approach outperforms attention-based models. An inspiring adjustment has been proposed by Bastings et al. (2019). They propose the use of the Kumaraswamy distribution (Kumaraswamy, 1980) instead of discrete selectors, which makes gradient estimation possible without REINFORCE.

We build on the work of Lei et al. (2016) and Bastings et al. (2019) and apply their approach to dialogue generation. Our work is different because neural dialogue generation systems present extra challenges for explainability. For example, neural dialogue systems have been shown to seldom understand or use the available dialog history effectively (Sankar et al., 2019). Also there are still no widely accepted metrics to judge the quality of the generated responses (Liu et al., 2016a), which makes it difficult to assess the quality of the rationales.

3 Methodology

The main setup is that we combine a language model with a rationale extractor model and train the combination on a joint objective to generate good responses while restricting the number of utterances or tokens that is used to generate that response. We experiment with two main variants: an embedding based rationale extractor and a policy-based rationale extractor. Figures 1 and 2 give an overview of how the language model is combined with the rationale extractor. The following paragraphs describe the three language models we used and the rationale extractors.

3.1 Language models

We combine the rationale extractors with three different language models: a basic LSTM, RoBERTa and DialoGPT.

The base language model we use is a simple LSTM with embedding size of 128 and two hidden layers with 128 nodes each. We use this LSTM to provide a baseline for our experiments.

RoBERTa (Liu et al., 2019) is a language model that builds on BERT (Devlin et al., 2018) and pretrains deep bidirectional representations from masking unlabeled text. It improves the pretraining procedure of BERT by changing key hyperparameters, removing the next-sentence training objective, dynamically changing the masking patterns and training with more data. Such modifications enable RoBERTa representations to generalize better to downstream tasks compared to BERT.

DialoGPT (Zhang et al., 2020) extends GPT-2 (de Vries and Nissim, 2020) to neural response generation tasks. GPT-2 is a 12-to-48 layer transformer-based language model, with generative pre-training on a diverse corpus of unlabeled text. Trained on very large datasets, it is able to generate text that is fluent, lexically diverse, and rich in content. DialoGPT is pre-trained on 147M conversation-like exchanges extracted from Reddit comment chains. It is the state-of-the-art model for dialogue response generation on multi-turn conversations. There are three sizes for pretrained model checkpoints: small (117M), medium (345M) and large (762M).

3.2 Rationale extractors

Two main types of rationale extractors are used, policy based and shared embedding based. The policy

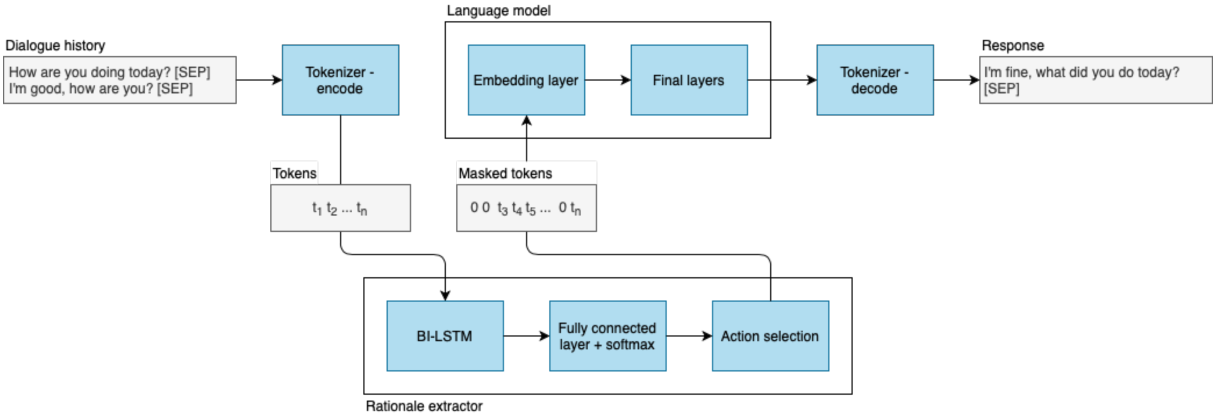

Figure 1: Model overview for policy based rationale extractor

based rationale extractor treats the language model as a black box and uses REINFORCE to optimize its parameters. The shared embedding based rationale extractor uses differential binary variables. For this the kumaraswamy distribution (Kumaraswamy, 1980) or the Gumbel-softmax trick (Jang et al., 2017) is used to make the model be end-to-end differentiable.

The policy based model has as input the tokens of the context, then for each token it decides whether it should mask it or not, see figure 1. This is done on based on a learned *policy* with a parametrization we denote with θ. The policy gives a probability of masking each token. To learn this policy we make use of the REINFORCE algorithm without a baseline.

Let z be a vector containing the rationale of length n, o the output of the language model and t the target tokens. We define the mask loss as:

$$m(z) = \lambda_0 \underbrace{\frac{1}{n}\sum_{i=1}^{n} z_i}_{L_0} + \lambda_1 \underbrace{\frac{1}{n}\sum_{i=1}^{n-1} |z_i - z_{i+1}|}_{\text{fussed lasso}}$$

In which λ_0 and λ_1 are hyperparameters. Let $L(0, t)$ be the cross entropy loss, the reward becomes:

$$r(o, t, z) = L(o, t) + m(z)$$

The loss for each sample $s = (o, t, z)$ is then given by:

$$L_{\text{sample}}(s|\theta) = -\sum_{i=1}^{n} r(o, t, z) \cdot \log(p(z_i|\theta))$$

The loss of a batch $b = (s^{(1)}, \ldots s^{(m)})$ becomes:

$$L_{\text{policy}}(b|\theta) = \frac{1}{m}\sum_{j=1}^{m} L_{\text{sample}}(s^{(j)}|\theta)$$

The parameters of the language models are also updated during the training of the rationale extractor. For this only the cross entropy loss is used.

The shared embedding models have as input the embedding of the tokens, see figure 2. For each embedded token it gets a weight. During training this can be a binary or any value on the interval $[0, 1]$. That is, both the Gumbel-softmax trick and Kumaraswamy allow to use values between 0 and 1 but can also be forced to be binarized.

These weights are then multiplied with the input embedding to get the output embedding which then is forwarded to the rest of the layers of the language model. As we have gradient approximation we can backpropagate through the whole model. As a loss we get:

$$L_{\text{shared}}(b) = \frac{1}{m}\sum_{j=1}^{m}(L(o^{(j)}, t^{(j)}) + m(z^{(j)}))$$

4 Experiments

In our experiments, we first train/finetune the language models, then we combine the language models with the rationale extractor and train both models concurrently.

Data For our experiments we select an open-domain dialogue dataset, DailyDialog (Li et al., 2017). We work with the DailyDialog dataset as all the information needed in the dialogue is in the dialogue itself and no additional information

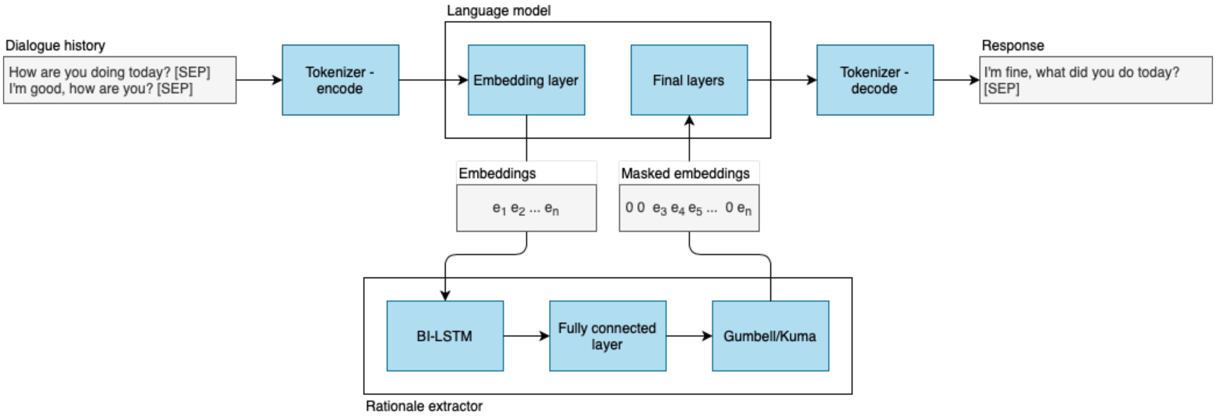

Figure 2: Model overview for embedding based rationale extractor

is needed (such as a knowledge graph). Furthermore each dialogue consists of multiple utterances per dialogue. This helps us to investigate the impact of the rationale extractor on dialogues of various lengths. The DailyDialog dataset consists of 13,118 multi-turn dialogues which are human written and reflect various topics about the daily life. The dataset is split into train/validation/test sets containing 11,118/1,000/1,000 dialogues respectively. For training, we preprocess the dataset and split each dialogue in a number of sub-dialogues. Each sub-dialogue consists of the context (multiple concatenated utterances) and the target response.

Language models The language models are trained based on the causal language prediction task with the help of teacher forcing. The hyperparameters of the language models are selected based on the validation loss. The basic language model (LSTM) is trained for 20 epochs with a learning rate of 0.001. For large language models, we finetune RoBERTa base and DialoGPT small models from the Huggingface library (Wolf et al., 2020). They are finetuned on DailyDialog with a learning rate of 1e-3 for 5 epochs and for 4 epochs respectively.

Rationale extractors For the *policy based* model the context is fed through the rationale extractor, we then concatenate the rationale with the target and forward it through the language model with the help of teacher forcing. We then compare the target with the output of the language model. The *shared embedding* models work similarly, but we use the embedded context instead of the context. As baseline models, we train the policy-based rationale extractors with randomly masking 10%, 25%, 50% of the tokens. We train the different combinations of rationale extractors and language models with a learning rate of 1e-4, a fussed lasso weight of 1e-3 and a sparsity weight of 0.01.

5 Results

5.1 Quantitative results

In quantitative analysis, we evaluate the performance of models on accuracy, perplexity and rationale, which is shown in Table 1.

Rationale extractor	Accuracy	Perplexity	Rationale
Combined with base LSTM as language model			
No rationale extractor	0.38	26	1.00
Policy based	0.37	30	0.42
Gumbel-Softmax	0.37	28	**0.15**
Kumaraswamy	0.37	28	0.50
Random 50%	0.37	29	0.50
Random 25%	0.36	30	0.25
Random 10%	0.36	31	0.25
Combined with DialoGPT as language model			
No rationale extractor	0.26	145	1.00
Policy based	0.29	86	0.71
Combined with roBERTa as language model			
No rationale extractor	0.06	357	1.00
Policy based	0.06	343	0.72

Table 1: Statistics of the models. Rationale is a real number between 0 and 1; 0 when the model is not using any of the tokens in the dialogue history, and 1 when it's using all tokens.

In models using the LSTM language model the rationale extractors can mask a lot of the context with a really small drop in both (top 1) accuracy and perplexity. However, the trained rationale extractors only seem marginally better than the random rationale extractors. From the trained models we see that Gumbel-softmax performs the best as it drops the most tokens with the smallest increase in perplexity.

In the experiments with the dialoGPT and

RoBERTa models, as we can see in Table 1, the language models perform better when combined with the policy rationale extractor. The perplexity is decreasing and the accuracy slightly goes up. Due to time constraints, we could not make the shared embedding rationale extractors work with large pre-trained language models.

Figures 3 and 4 show graphs with the relative positions of the tokens in the context for the LSTM model respectively for the dialoGPT and RoBERTa models. Here 0 is the most recent and 100 indicates the start of the context. Both the Kumaraswamy and the Gumbel rationale extractor mostly select tokens near the start and the end of the dialogue history. This is in contrast to the policy and the random distance distribution which select tokens nearly uniformly spread across the context. The policy based model is virtually indistinguishable from the random distribution.

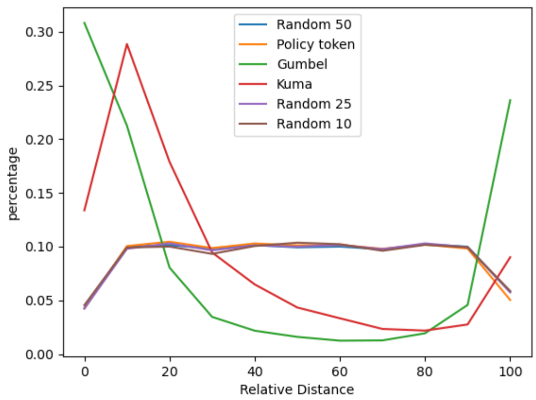

Figure 3: Relative position of used tokens in LSTM language model (0=most recent token, 100=oldest token)

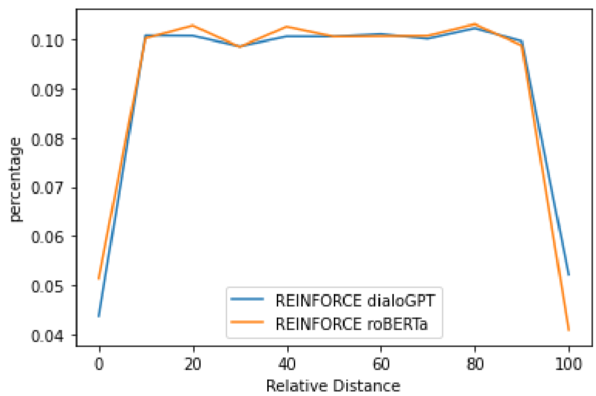

Figure 4: Relative position of used tokens in DialoGPT & RoBERTa (0=most recent token, 100=oldest token)

5.2 Qualitative results

When we look at figure 5 we see that the generated responses are locally consistent: in general they form coherent sentences. But the generated dialogue is non-sensible. The quality between models does not seem to differ. We did not managed to get RoBERTa to generate any sensible dialogue at all as it only seems to generate separate tokens.

6 Analysis and discussion

The results provide a first look at the potential of using rationale extractors for neural dialogue models. In this section we highlight some interesting findings and discuss how one could improve upon the results. Due to the limited time of the project our research has also some shortcomings, which we also mention. We conclude with suggestions for future work. We start by looking at how the model make use of the history, then we compare the shared and policy based rationale extractors. Lastly we discuss potential problems with the learning, shortcomings and suggestions for future work.

6.1 Effective use of the history

In general we see that the rationale extraction does not seem to harm the performance of the models that much. For example looking at the Gumbell model we see that with only 15% of the tokens we still can make almost as good as predictions as before. Furthermore we see that the shared embedding model mainly makes use of the last part of the contexts and from the start but not really the parts in between which ties in with the findings of (Sankar et al., 2019)). Difference is that our setup should, in principle, learn which parts of the context can be masked.

Using the start and the end of a conversation seems sensible, because the topic of a conversation is often determined at the start, and the next response should be consistent with that topic and with the last utterance.

Interesting is the difference in the length of the rationale extracted between the Gumbell and the Kumaraswamy model. We do not have a good explanation for the difference between these model and did not find research that compares the Gumbell-Softmax trick and the Kumaraswamy method.

Models	Dialog history	Response
Policy+LSTM	"how are you doing ? [SEP]	i ' m not sure , but why are you going to get something from me ? [SEP]
	"how are you doing ? [SEP] i ' m not sure , but why are you going to get something from me ?	you can do something to drink . but i ' d like to play a few things i have been [SEP]
Gumbel+LSTM	how are you doing ? [SEP] i was in the class , but you were really noticed me . [SEP]	i am not going to be a good job. [SEP]
	how are you doing ? [SEP] i was in the class , but you were really noticed me . [SEP] i am not going to be a good job . [SEP]	what ? why are you going to work in ? [SEP]
Kuma+LSTM	how are you doing ? [SEP] i ' m going to the doctor for my friends . [SEP]	what are your plans for the job ? [SEP]
	how are you doing ? [SEP] i ' m going to the doctor for my friends . [SEP] what are your plans for the job ? [SEP]	i want to know about the same time . [SEP]
Random 50%+LSTM	how are you doing ? [SEP] fine , thanks . i am afraid of my friend . [SEP]	what ' s that ? [SEP]
	how are you doing ? [SEP] fine , thanks . i am afraid of my friend . [SEP] what ' s that ? [SEP]	what do you think of it ? [SEP]
Policy+DialoGPT	how are you doing? [SEP] Well . [SEP].	Alright. [SEP]
	how are you doing? [SEP]Well. [SEP]Alright. [SEP]	You're welcome. But they are you want to get some things here for you know I'm afraid[SEP]
Policy+Roberta	<s>how are you doing? [SEP]</s>[SEP]	<s>..[SEP]
	<s>how are you doing? [SEP]</s>[SEP] <s>..[SEP]	</s> [SEP]

Figure 5: Rationale(in blue) examples of different combinations of language models and rationale extractors

6.2 Difference between shared and policy

Although the policy rationale extractor did not seem to work there are still good reasons to investigate the policy based models further. One of the reasons is that a policy could integrate explicit exploration during training. All the differences between the shared and policy based RE are highlighted in the table 2

Policy	Embedding
LM as blackbox	LM as grey box
Gradient estimate	Gradient estimate or true gradient
Training with reinforcement algorithms	Gradient descent only
Able to explore	No explicit way to explore

Table 2: The differences between policy based and shared embedding rationale extractor

6.3 Learning problems

Reward assignment problem The policy based model seems to select randomly from the tokens. This could be due to the reward assignment problem. This problem arises when it is hard to find which of the taken action has the biggest impact on the reward. As masking or not masking a token is one action, masking the context is based on as many actions as there are tokens. To find out which (combination) of actions leads to the best rewards is therefore hard and doing random masking may seem to a the strategy that the model converges to. Other learning objects that lead to more informed gradients may tackle this problem. For example (Yu et al., 2019) introduces an adversarial setting in which leaving important information out of the rationale is directly penalized, which leads to better rationals and a more informative reward signal.

Teacher forcing During the training we make use of teacher forcing. The problem with teacher forcing is that it gives actual token after the model generated its predicted token. This prevents drift in the error during training, but it also gives the language model information that it can use to predict the next token. In theory the model could get the first few tokens wrong but then, based on the tokens provided by teacher forcing, still predict the next tokens fairly accurate. Other training methods that do not make use of teacher forcing such as described in (Liu et al., 2016b) could be used to tackle this problem.

6.4 Shortcomings

Better language models The language model we train with shared embedding rationale extractor is a 2-layer pretrained LSTM model, which does not perform very well in dialogue generation. Such issue might limit the rationale performance. Though attempts have been made to combine the shared embedding RE with pretrained large-scale language models, we didn't make the model work properly due to technical problems on our part with the pre-trained (huggingface.co) models.

Metrics Due to time constraints we could only collect measurements about accuracy, perplexity and number/ position of the used tokens. These metrics are not sufficient to judge the quality of the rationales. We propose to measure the quality in terms of faithfulness and plausibility. *Faithfulness* means that changing the rationale (e.g. dropping an extra word, or selecting different words) would harm the quality of the response. A faithful explanation is both sufficient (adding more words does not yield better results) and necessary (removing words from the explanation does yield a worse result). This can be measures comparing the performance of the model when we use a perturbed rationale instead of the actual rationale. A rationale can be considered *plausible* if it provides a human-understandable justification for the model's predictions. This requires assessment of the rationales by human assessors.

RoBERTa poor performance The RoBERTa model combined with the policy rationale extractor is failing to generate any meaningful responses. In addition, it's performing very poorly in terms of accuracy and perplexity (see table 1). That is probably, due to implementation issues that we could not resolve in time.

6.5 Future work

The current paper compares rationale extractor models on a single open-domain dataset. We think it is interesting to compare the rationales for open domain dialogue systems with those for task oriented dialogue systems. Additional analysis can be done to assess the quality (sufficiency and necessity) of the generated rationals by incorporating perturbations on the data. Furthermore it would be interesting to combine the shared embedding rationale extractor with better language models such as RoBERTa. Lastly, experiments with different training setups and different loss functions could lead to insights in how one could train a rationale extractor for dialogue modelling.

7 Conclusion

Our experiments are a first attempt to use a rationale extractor to improve explainability of neural dialog generation models. Although our initial results are limited, our experiments provide a good basis for further research by providing a practical implementation and insights into the barriers that still have to be overcome to realise faithful and plausible explanations.

Acknowledgments

We would like to express our gratitude to Raquel Fernández, Mario Giulianelli and Ece Takmaz for offering their guidance and feedback during this research project. In addition, we would like thank the SURF.nl organization which offered us free GPU resources to conduct our experiments. Last but not least, we are thankful to all the fellow students who provided their feedback and helped us to improve this project.

References

Joost Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

P. Kumaraswamy. 1980. A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1):79–88.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016a. 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*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, Jifan Chen, and Xuanjing Huang. 2016b. Deep fusion LSTMs for text semantic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1034–1043, Berlin, Germany. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Chinnadhurai Sankar, Sandeep Subramanian, Chris Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do Neural Dialog Systems Use the Conversation History Effectively? An Empirical Study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 32–37, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wietse de Vries and Malvina Nissim. 2020. As good as new. how to successfully recycle english gpt-2 to make models for other languages.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. 2019. Rethinking cooperative rationalization: Introspective extraction and complement control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4085–4094.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation.