Beyond User Self-Reported Likert Scale Ratings: A Comparison Model for Automatic Dialog Evaluation

Weixin Liang[1], **James Zou**[1] **and Zhou Yu**[2]

[1] Stanford University,[2] University of California, Davis
{wxliang, jamesz}@stanford.edu, joyu@ucdavis.edu

Abstract

Open Domain dialog system evaluation is one of the most important challenges in dialog research. Existing automatic evaluation metrics, such as BLEU are mostly reference-based. They calculate the difference between the generated response and a limited number of available references. Likert-score based self-reported user rating is widely adopted by social conversational systems, such as Amazon Alexa Prize chatbots. However, self-reported user rating suffers from bias and variance among different users. To alleviate this problem, we formulate dialog evaluation as a comparison task. We also propose an automatic evaluation model CMADE (Comparison Model for Automatic Dialog Evaluation) that automatically cleans self-reported user ratings as it trains on them. Specifically, we first use a self-supervised method to learn better dialog feature representation, and then use KNN and Shapley to remove confusing samples. Our experiments show that CMADE achieves 89.2% accuracy in the dialog comparison task. Our implementation is available at `https://github.com/Weixin-Liang/dialog_evaluation_CMADE`.

1 Introduction

Open-domain dialog system evaluation is one of the most difficult challenges in the dialog community. Open-domain chatbots have a user-centric goal: to provide human with enjoyable user experience. However, user experience is difficult to quantify due to bias and variance among different users. Previous research has optimized on automatic dialog evaluation metrics such as BLUE (Papineni et al., 2002), which measures the difference between the generated responses and the reference responses. Due to the contrast between the one-to-many nature of open-domain conversations and the limited number of available references, such metrics correlate poorly with human judgments (Liu et al., 2016; Lowe et al., 2017; Novikova et al., 2017). Designing a fully automatic dialog evaluation metric is still an open research problem.

Currently, both academia and industry (Ram et al., 2018a; Li et al., 2019b; Liang et al., 2019) rely on human ratings to evaluate open-domain dialogs. Following the ubiquitous application of Likert scores in survey research like online reviews (Godes and Silva, 2012) and consumer satisfaction (Peterson and Wilson, 1992), a common practice of human evaluation on dialogs is to ask either a third-person rater or the chatbot user to report a Likert score. However, concerns have been raised about the validity of Likert score-based ratings. Kulikov et al. (Kulikov et al., 2018) observe high bias and variance of Likert scores. Such issue is more severe in real-world commercial dialog systems like Alexa social chatbot (Ram et al., 2018a; Venkatesh et al., 2018), because the real-world users have neither monetary incentive nor necessary annotation training to calibrate their ratings.

To explore the validity of Likert score based dialog evaluation, we first perform a large-scale data analysis of 3,608 collected real-world human-machine dialogs along with their self-reported Likert scale ratings from Amazon Alexa Prize Challenge (Ram et al., 2018a; Yu et al., 2019; Chen et al., 2018). One noticeable property of the ratings is its J-shape skew distribution: nearly half of the dialogs are rated with the highest Likert score. The prevalence of such extreme distribution of ratings has long been observed by the business research community in variable aspects of real-life (Schoenmüller et al., 2018; Godes and Silva, 2012; Hu et al., 2017; Zervas et al., 2015).

Although we could tell which dialog system is better by running statistical test on a large number of noisy ratings, it is difficult to locate dialogs with bad performance reliably to improve dialog system quality. In this paper, we take on the challenge of calibrating a large number of noisy self-reported user ratings to build better dialog evaluation models. We formulate the task as to first denoise the self-reported user ratings and then train a model on the cleaned ratings. We design CMADE (Comparison Model for Automatic Dialog Evaluation), a progressive three-stage denoising pipeline. We first perform a self-supervised learning to obtain good dialog representations. We then fine-tune CMADE on smoothed self-reported user ratings to improve the dialog representation while preventing the network from overfitting on noisy ratings. Finally, we apply data Shapley to remove noisy training data, and fine-tune the model on the cleaned training set. Our experiments show that CMADE is able to successfully identify noisy training data and achieves 89.2% in accuracy and 0.787 in Kappa on a test set with unseen expert-rated dialog pairs.

2 Related Work

Open-domain dialog system evaluation is a long-lasting challenge. It has been shown that previous automatic dialog evaluation metrics correlate poorly with human judgments (Liu et al., 2016; Lowe et al., 2017; Novikova et al., 2017). A well-known reason is that these automatic dialog evaluation metrics rely on modeling the distance between the generated response and a limited number of references available. The fundamental gap between the open-ended nature of the conversations and the limited references (Gupta et al., 2019) is not addressed in methods that are lexical-level based (Papineni et al., 2002; Lin, 2004; Banerjee and Lavie, 2005), embedding based (Rus and Lintean, 2012; Forgues et al., 2014), or learning based (Tao et al., 2018; Lowe et al., 2017).

Given the aforementioned limitations, Likert-score based rating is the de-facto standard for current dialog research and social conversational systems such as in Amazon Alexa Prize Challenge (Yu et al., 2019; Chen et al., 2018). Various forms of evaluation settings have been explored to better measure human judgments. Single-turn pairwise comparison (Vinyals and Le, 2015; Li et al., 2016) is primarily used for comparing two dialog systems. Each system predicts a single utterance given the static "gold" context utterance from human-human logs. Although such A/B test setting is robust to annotator score bias, it cannot capture the multi-turn nature of dialogs. A more complete multi-turn evaluation is typically measured with a Likert scale for the full dialog history, where either a third-person rater or the chatbot user (Pérez-Rosas et al., 2019) reports a Likert score on user experience (Venkatesh et al., 2018), engagement (Bohus and Horvitz, 2009) or appropriateness (Lowe et al., 2017). However, as observed in (Kulikov et al., 2018; Ram et al., 2018a; Venkatesh et al., 2018) Likert scores suffer from bias and variance among different users. Different from previous empirical observations, we conduct a large-scale quantitative and qualitative data analysis of Likert score based ratings. To address the issue of Likert scores, the Alexa team proposed a rule-based ensemble of turn-granularity expert ratings (Yi et al., 2019), and automatic metrics like topical diversity (Guo et al., 2018) and conversational breadth. ACUTE-EVAL (Li et al., 2019a) makes a small-scale attempt to use multi-turn pair-wise comparison to rank different chatbots. Given the ubiquity and simplicity of Likert scores based evaluation, instead of proposing an alternative measure, we take on the challenge of denoising Likert scores with minimal expert annotations introduced (one order of magnitude smaller). Different from (Li et al., 2019a), our proposed expert annotation scheme is for comparing the dialogs within the same chatbot.

3 Dialog Rating Study

3.1 Dataset

The data used in this study was collected during the 2018 Amazon Alexa Prize Competition (Ram et al., 2018b). Our data contain long and engaging spoken conversations between thousands of real-world Amazon Alexa customers and Gunrock, the 2018 Alexa Prize winning social bot (Yu et al., 2019). The chatbot has 11 topic dialog modules including movies, books, and animals. One notable characteristic of the chatbot is its versatile and complex dialog flows which interleaves facts, opinions and questions to make the conversation flexible and interesting (Chen et al., 2018). At the end of each dialog, a self-reported Likert scale rating is elicited by the question "on a scale of one to five, how likely would you talk to this social bot again?"

We first filter out dialogs that have inappropriate content using keyword matching. We then select 3,608 ten-turn dialogs on movies, because movie dialogs are more coherent and diverse compared to other topics according to both real users and Amazon selected experts. We observe that dialogs

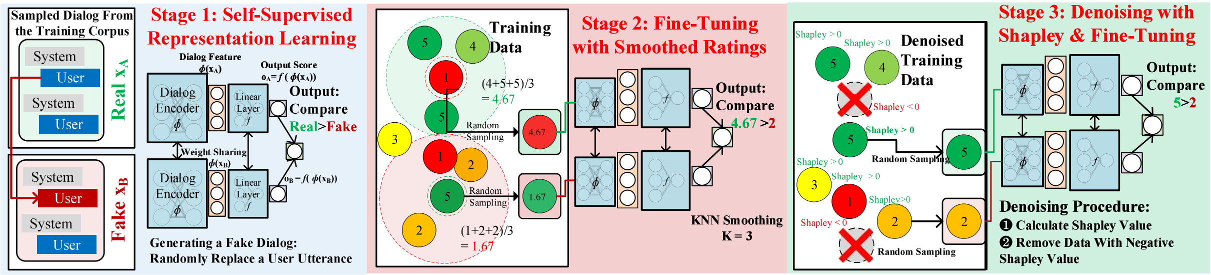

Figure 1: Schematic of the CMADE workflow. CMADE contains a three-stage training pipeline to denoise self-reported ratings to train an automatic dialog comparison model: learning representation viaself-supervised dialog flow anomaly detection, fine-tuning with smoothed self-reported user ratings, denoising with data Shapley & further fine-tuning. The gray and blue rectangles in stage 1 represents system and user utterances. The red rectangle in stage 1 represents the randomly replaced system utterance for dialog flow perturbation. In stage 2 & 3, each ball represents a dialog in the training data. The number on each ball represents the dialog rating.

with more than eight turns are more meaningful and semantically versatile, while dialogs more than 10 turns exceed the max length limit of the BERT model (512 tokens). So we select dialogs that have ten turns. Our approach could support longer conversations by adopting a memory footprint efficient algorithm for self-attention to support sequences with thousands of tokens (Huang et al., 2019). We leave this to future work.

We aim to evaluate user experience for each dialog from the same chatbot of the same length. This is significantly more challenging than identifying which chatbot provides a better user experience on average since our problem setup requires us to capture more subtle difference in user experience.

3.2 Likert Score Based Evaluation

Rating	1	2	3	4	5
Count	386	404	566	664	1588
Fraction	10.7%	11.2%	15.7%	18.4%	44.0%

Table 1: The statistics of self-reported Likert scale ratings. The distribution is heavily skewed and noisy: nearly half of the dialogs are rated with score = 5.

J-Shape Skewness We perform a detailed analysis of the self-reported Likert scale ratings. As shown in Table 1, abnormally, nearly half of the dialogs are rated as five, which is the highest score. A similar skewed distribution is also observed in previous years' Alexa competition (Fang et al., 2018). In fact, the business research community has long observed the prevalence of the extreme distribution of reviews in which the reviews are heavily skewed to the positive end of the rating scale (known as ”J-shape”) in online reviews (e.g., Amazon, Airbnb, Yelp) (Godes and Silva, 2012; Hu et al., 2017; Zervas et al., 2015), word of mouth (East et al., 2007) and consumer satisfaction (Peterson and Wilson, 1992; Danaher and Haddrell, 1996).

Comparison to expert ratings We randomly selected 50 dialogs rated score-5 and showed these to an expert, and our expert rated 27 of them with score-4 or less. The Alexa team (Venkatesh et al., 2018) has also reported that the inter-user agreement is quite low for their internal rating analysis. Such phenomena indicate that the self-reported Likert scale ratings are extremely noisy. Using such ratings cannot localize individual bad interactions. In addition, Likert score based evaluation also suffers from insensitivity issues. As observed by the Alexa team (Venkatesh et al., 2018) in multiple internal user studies, even though users evaluated multiple dialogs with the same score, they had a clear rank order among the dialogs.

The skewness, noisiness and insensitivity of the self-reported Likert scale rating make it a sub-optimal dialog evaluation metric. In practice, we find that directly training a classifier (even for pre-trained BERT-based model) on the noisy self-reported Likert scale ratings suffers from underfitting. One of the Alexa Price Challenge team, Alana (Papaioannou et al., 2017) train a binary-classifier between successful dialogs (human rating 4 or 5) and unsuccessful dialogs (rating 1 or 2) with heavy hand-engineered features. They reach 69.40% accuracy on this binary classification problem, which is far from usable in real-world settings.

3.3 Pairwise Comparison Based Evaluation

Selecting the better dialog from two options is easier for a human evaluator than giving an absolute number like the Likert score, which requires the evaluator to maintain a consistent standard. People's perception is inherently relative, and pair-wise comparison is local and does not require the user to have global consistency. There are many other examples where humans find it easier to perform pairwise comparisons rather than providing direct labels (Simpson and Gurevych, 2018; Mailthody et al., 2019; Liang et al., 2018), including content search (Fürnkranz and Hüllermeier, 2010), image retrieval (Wah et al., 2014; Feng et al., 2019), and age estimation (Zhang et al., 2017).

We randomly sample 400 dialog pairs for experts to annotate. We ask the question, "If you were the user, in which scenario would you be more likely to come back and talk to the system again? " We guide the experts to focus on the user experience rather than calibrating the performance of any specific module of the dialog system. Two researchers with conversational training experience annotated the data. The leading expert has been working in an Alexa competition team for more than one year with an emphasis on the user ratings. For each dialog pair (A, B), they label 'A is better than B' or 'B is better than A' or 'cannot tell'. They reached a high inter-annotator agreement score (Cohen, 1968) with kappa $\kappa = 0.83$. To make sure that the dev & test is accurate, we throw away all "cannot tell" dialog pairs. We then study the correlation between Likert score based evaluation and pairwise comparison based evaluation.

3.4 Correlation Between User Ratings and Expert Ratings

Delta of Self-Reported Ratings (e.g., 5-1=4)	Δ=1	Δ=2	Δ=3	Δ=4
Disagreement Rate	0.45	0.383	0.220	0.157

Table 2: The correlation between the self-reported Likert scale ratings and our pair-wise comparison annotation. For a pair of dialogs, if the delta of self-reported Likert scale ratings is large, then they are more likely to align with the comparison results from experts.

To further analyze the self-reported Likert scale ratings, we also compare the annotated labels of the 403 dialog pairs with the self-reported Likert scale ratings of these dialogs. For each pair of dialogs, we compare the pairwise comparison label and the delta between the self-reported Likert scale ratings of the two dialogs. Ideally, the dialog with a higher self-reported Likert scale rating should be the one that is annotated as having a better user experience in the pairwise comparison. We count the number and fraction of "disagreement" between the two types of ratings. Overall, roughly 1/3 of the dialog pairs disagree. As shown in Table 2, as the gap between the self-reported Likert scale ratings becomes larger, the disagreement between expert and self-reported ratings goes down. This suggests that if the difference between the two dialogs' Likert score is huge, they are more likely to be consistent with the comparison ratings.

4 Problem Formulation

Suppose the training set D_{train} consists of data points $D_{train} = \{(x_i, y_i)\}_1^{N_{train}}$ where x_i is a dialog and y_i is the noisy self-reported user ratings. We define a strict partial order relationship $\triangleright$ where $x_i \triangleright x_j$ means that dialog x_i provides a better user experience than dialog x_j. Note that $y_i > y_j$ does not always imply $x_i \triangleright x_j$ since self-reported user ratings are noisy (§ 3.3, § 3.4). The test set D_{test} consists of N_{test} dialog pairs along with their binary pair-wise comparison labels $D_{test} = \{(x_i^{test}, x_j^{test}, z_{i,j}^{test})\}_{i,j \in I^{test}}$, where $z_{i,j}^{test}$ is annotated by experts and indicates whether dialog A provides a better user experience than dialog B, i.e., $z_{i,j}^{test} = \mathbb{1}(x_i \triangleright x_j)$. The development set D_{dev} has a similar structure.

Following the structure of the expert annotated pairs, we formulate our model $M(\phi, f)$ as a pair-wise dialog predictor with a similar architecture as RankNet (Burges et al., 2005). For a dialog pair (x_i, x_j), the model predicts an un-normalized score $o_i, o_j \in \mathbb{R}$ for each dialog: $o_i = f(\phi(x_i))$ and $o_i = f(\phi(x_j))$ where ϕ is a dialog encoder that maps each dialog to a feature space and f is a linear transformation that converts each dialog feature into a real number o. We define a binary relationship $\hat{\triangleright}$ where $x_i \hat{\triangleright} x_j$ means that the model predicts that dialog x_i provides a better user experience than dialog x_j. We denote model's prediction of $z_{i,j}$ as $\hat{z_{i,j}}$ where $\hat{z_{i,j}} = \mathbb{1}(x_i \hat{\triangleright} x_j)$. We model the predicted posterior $P(\hat{z_{i,j}} = 1) = P(x_i \hat{\triangleright} x_j)$ as:

$$P(\hat{z_{i,j}} = 1) = P(x_i \hat{\triangleright} x_j) = \frac{1}{1 + e^{-(o_i - o_j)}}$$

5 Method

Our goal is to reduce the noise of the self-reported user ratings (§ 3). Directly training a classification model using the noisy ratings leads to severe underfitting. To this end, we propose a three-stage training pipeline to denoise self-reported ratings to train an automatic dialog comparison model. Figure 1 describes the overall pipeline:

- In Stage 1, we learn dialog feature representation with a self-supervised dialog flow anomaly detection task.
- In Stage 2, we perform label smoothing to adjust the noisy self-reported ratings in the training set and fine-tune the dialog comparison model on the smoothed ratings.
- In Stage 3, we perform data Shapley (Ghorbani and Zou, 2019; Jia et al., 2019a) on the self-reported user ratings to identify and remove noisy data points. We then fine-tune the dialog comparison model on the cleaned training set.

5.1 Stage 1: Learning Representation via self-supervised dialog anomaly detection

Sys: What movie did you see? **User:** Spider man into the spider verse
Sys: Ah, I know about Spider man into the spider verse! I'm wondering. What would you rate this movie on a scale from 1 to 10?
Replaced Sys: Isn't it crazy how famous actors can get? Are you interested in talking more about Scarlett Johansson?

Table 3: A fake dialog example created by dialog flow perturbation in Stage 1. We perturb the dialog flow by replacing a system utterance (here the second Sys utterance in the table) with a random system utterance from the corpus (here the replaces Sys utterance) to generate a fake dialog. With high probability, the fake dialog is less appropriate than the origin one.

Having a good dialog representation is the first step towards denoising the data. Our primary goal in this stage is to train a dialog encoder ϕ to learn good dialog feature representations for the following stages. Here ϕ could be any sequence encoder that could encode a dialog and we use BERT (Devlin et al., 2019) in this paper.

For each dialog in the training set, we perturb the dialog flow to generate a fake dialog and train the model to differentiate the fake dialog and the real one. Dialog flow is a user-centric measure of whether a conversation is "going smoothly" (Eskénazi et al., 2019). To perturb the dialog flow for each dialog x_i, we randomly replace a user utterance in x_i with a random user utterance from the training corpus D_{train}, yielding a perturbed dialog $x_{i,fake}$. With high probability, the system utterance immediately following the replaced user utterance becomes inappropriate. Therefore, we incorporate $\{(x_i, x_{i,fake}, z = 1)\}$ into the training pairs. Similarly, we also randomly replace a system utterance and yield another perturbed dialog. We generate two perturbed dialogs for each dialog in the training set and thus $2N_{train}$ real-fake dialog pairs in total. An example is shown in Table 3. We note that appropriateness is one of the most widely applied metrics of human evaluation on dialogs (Lowe et al., 2017). By learning to differentiate the perturbed dialog and the original one, we expect CMADE to learn a good dialog encoder ϕ which maps dialogs with similar dialog flow close to each other in the feature space.

5.2 Stage 2: Fine-tuning with smoothed self-reported user ratings

Stage 1 only performs unsupervised learning and does not incorporate any supervision from human ratings. To obtain better dialog feature representations for Stage 3, Stage 2 fine-tunes ϕ with supervision from the noisy self-reported user ratings. We adopt a simple yet effective label smoothing, inspired by (Szegedy et al., 2016; Nie et al., 2019), using the representation learned in Stage 1. A key assumption in Stage 2 is that dialogs with similar dialog flow provide a similar user experience. For each dialog x_i, we find its K nearest neighbors in the feature space defined by ϕ. We use the average self-reported ratings of the K nearest neighbors as a smoothed rating y_i^s for x_i. To construct training dialog pairs, we randomly sample dialog pairs x_i and x_j and derive a pair-wise comparison label $z_{i,j}^s$ by comparing the smoothed rating y_i^s and y_j^s: $z_{i,j}^s = \mathbb{1}(y_i^s > y_j^s)$. We discard the pairs with equal y_i^s and y_j^s. To improve the dialog feature representation, we fine-tune the model $M(\phi, f)$ on sampled dialog pairs along with the derived labels from comparing the smoothed scores $\{x_i, x_j, z_{i,j}^s\}$. We note that $z_{i,j}^s$ depends solely on the noisy self-reported ratings in the training set and does not depend on the expert annotations. Theoretically, we could iterate between label smoothing and model fine-tuning since the fine-tuned model provides better dialog feature representation. In practice, we find that one iteration is enough to reach good prediction performance.

Label smoothing has led to state-of-the-art models in image classification (Szegedy et al., 2016), language translation (Vaswani et al., 2017) and speech recognition (Chorowski and Jaitly, 2017). Prior attempts in label smoothing (Szegedy et al., 2016; Vaswani et al., 2017; Chorowski and Jaitly, 2017; Müller et al., 2019) focus on categorical labels to prevent the network from becoming overconfident while we apply label smoothing on ordinal labels (i.e., Likert scores) to prevent the network from overfitting on noisy ordinal labels.

5.3 Stage 3: Denoising with data Shapley & further fine-tuning

In Stage 2, noisy ratings still have effect in the smoothed ratings for other data points. In Stage 3, we aim to identify and remove dialogs with noisy self-reported user ratings y_i with data Shapley value technique (Ghorbani and Zou, 2019; Jia et al., 2019a,b). Shapley value comes originally from cooperative game theory (Dubey, 1975). In a cooperative game, there are n players $D = \{1, ..., n\}$ and a utility function $v : 2^{[n]} \to \mathbb{R}$ assigns a reward to each of 2^n subsets of players: $v(S)$ is the reward if the players in subset $S \subseteq D$ cooperate. Shapley value defines a unique scheme to distribute the total gains generated by the coalition of all players $v(D)$ with a set of appealing mathematical properties. Shapley value has been applied to problems in various domains, ranging from economics (Gul, 1989) to machine learning (Cohen et al., 2005; Yona et al., 2019).

In our setting, given $D_{train} = \{(x_i, y_i)\}_1^{N_{train}}$, we view them as N_{train} players. We could also view the utility function $v(S)$ as the performance on the development set. The Shapley value for player i is defined as the average marginal contribution of $\{(x_i, y_i)\}$ to all possible subsets that are formed by other users (Jia et al., 2019a):

$$s_i = \frac{1}{N} \sum_{S \subseteq D_{train} \backslash \{x_i\}} \frac{1}{\binom{N-1}{|S|}} [v(S \cup \{x_i\}) - v(S)]$$

As suggested by the definition of data Shapley, computing data Shapley value requires an exponentially large number of computations to enumerate $\mathcal{O}(2^{N_{train}})$ possible subsets and train the model M on each subset, which is intractable. Inspired by (Jia et al., 2019a), CMADE tackles this issue by reducing the deep model M to a k-nearest neighbors (KNN) model and then apply the closed-form solution of shapley value on KNN. Using the feature extractor ϕ trained in Stage 1 and Stage 2, we fix ϕ and map all dialogs in the training data $\{x_i\}_1^{N_{train}}$ to $\{\phi(x_i)\}_1^{N_{train}}$. We first define the utility function $v(S)$ in a special case where the development set only contains one dialog pair $(x_p^{dev}, x_q^{dev}, z_{p,q}^{dev})_{p,q \in I^{dev} = \{(1,2)\}}$. In our setting, the development set contains dialog pairs annotated by experts. Given any nonempty subset $S \subseteq D_{train}$, we use the KNN Regressor to rate x_p^{dev} and x_q^{dev}. To do this, we compute $\phi(x_p^{dev})$ and sort $\{x_p\}_1^{N_{train}}$ based on their euclidean distance in the dialog feature space to x_p^{dev}, yielding $(x_{\alpha_1^{(p)}}, x_{\alpha_2^{(p)}}, ..., x_{\alpha_{|S|}^{(p)}})$ with $x_{\alpha_1^{(p)}}, ..., x_{\alpha_K^{(p)}}$ as the top-K most similar dialogs to x_p^{dev}. Similarly, we get $(x_{\alpha_1^{(q)}}, x_{\alpha_2^{(q)}}, ..., x_{\alpha_{|S|}^{(q)}})$ with $x_{\alpha_1^{(q)}}, ..., x_{\alpha_K^{(q)}}$ as the top-K most similar dialogs to x_q^{dev}. Based on the self-reported user ratings in the training data, we use the KNN Regressor to rate x_p^{dev} and x_q^{dev} as follows:

$$\hat{y}_p^{dev} = \frac{1}{K} \sum_{k=1}^{min\{K,|S|\}} y_{\alpha_k^{(p)}} \tag{1}$$

$$\hat{y}_q^{dev} = \frac{1}{K} \sum_{k=1}^{min\{K,|S|\}} y_{\alpha_k^{(q)}} \tag{2}$$

The model predicts $\hat{z}_{p,q}^{dev} = 1$ if $\hat{y}_p^{dev} > \hat{y}_q^{dev}$ and vice versa.

To obtain a closed-form solution to calculate Shapley value, instead of defining the utility function as the accuracy of the pair-wise prediction, we define the utility function as follows:

$$v(S) = \begin{cases} \hat{y}_p^{dev} - \hat{y}_q^{dev}, & \text{if} \quad z_{p,q}^{dev} = 1, \\ \hat{y}_q^{dev} - \hat{y}_q^{dev}, & \text{if} \quad z_{p,q}^{dev} = 0 \end{cases} \tag{3}$$

Theorem 1 *Consider the utility function in Equation (3). Then the Shapley value of each training point s_m can be decomposed into two terms $s_m^{(p)}$ and $s_m^{(q)}$ which depend on x_p^{dev} and x_q^{dev} respectively. $s_m^{(p)}$ and $s_m^{(q)}$ can be calculated recursively as follows:*

$$s_m = \begin{cases} s_m^{(p)} - s_m^{(q)}, & \text{if} \quad z_{p,q}^{dev} = 1, \\ s_m^{(q)} - s_m^{(p)}, & \text{if} \quad z_{p,q}^{dev} = 0 \end{cases}$$

$$s^{(p)}_{\alpha_N^{(p)}} = \frac{y_{\alpha_N^{(p)}}}{N} \qquad s^{(q)}_{\alpha_N^{(q)}} = \frac{y_{\alpha_N^{(q)}}}{N}$$

$$s^{(p)}_{\alpha_m} = s^{(p)}_{\alpha_{m+1}^{(p)}} + \frac{y_{\alpha_m^{(p)}} - y_{\alpha_{m+1}^{(p)}}}{K} \frac{min\{K, m\}}{m}$$

$$s^{(q)}_{\alpha_m^{(q)}} = s^{(q)}_{\alpha_{m+1}^{(q)}} + \frac{y_{\alpha_m^{(q)}} - y_{\alpha_{m+1}^{(q)}}}{K} \frac{min\{K, m\}}{m}$$

With Theorem 1, the Shapley value calculation could be finished in $\mathcal{O}(N \log N)$ time. The above result for a single point in the development set could be readily extended to the multiple-testpoint case. In our experiment, with such optimization, the Shapley value calculation takes less than 5 seconds to finish. Theorem 1 comes primarily from (Jia et al., 2019a,b) and we extends their results of vanilla KNN regressor (Jia et al., 2019a) to our pairwise testing setting.

By applying the Shapley technique to the data, we identify noisy training data points which contribute negatively to the performance and remove them from the training set. Similar to Stage 2, to construct training dialog pairs, we randomly sample dialog pairs x_i and x_j from the cleaned training set and derive $z_{i,j}$ by comparing the self-reported rating y_i and y_j. We then further fine tune the model from Stage 2. Theoretically, we could iterate between Stage 2 and Stage 3 multiple times while in practice one iteration is enough.

5.4 Towards Scalable Pair-based Training

We use a similar factorization technique for pairwise ranking in LambdaRank (Burges et al., 2006) to speed up training. For Stage 2 and 3, we have $\mathcal{O}(N^2)$ possible dialog pairs, which leads to quadratically increasing training time. Similar to LambdaRank (Burges et al., 2006), it is possible to calculate the exact gradient of $\mathcal{O}(N^2)$ possible dialog pairs with $\mathcal{O}(N)$ forwards and back-propagations. More specifically, we denote the possible input pairs during training at Stage 2 or Stage 3 as: $D_{train}^{pair} = \{(x_i, x_j, z_{i,j})\}_{i,j \in I}$. The total cost L for $\mathcal{O}(N^2)$ possible dialog pairs is the sum of $\mathcal{O}(N^2)$ cross-entropy costs:

$$L_{i,j} = CrossEntropy(\hat{z}_{i,j}, z_{i,j})$$
$$L = \sum_{(i,j) \in I} L_{i,j}$$

Theorem 2 *We can compute* $\frac{\partial L}{\partial w_k}$ *in* $\mathcal{O}(N)$ *by factor it into a weighted sum of* $\frac{\partial o_i}{\partial w_k}$ *where the weight* $\lambda_i \in \mathbb{R}$ *only depends on* $\{o_j\}$ *and* $\{z_{i,j}\}$*. W.l.o.g., we assume* $z_{i,j} \equiv 1$ *.*

$$\frac{\partial L}{\partial w_k} = \sum_i \lambda_i \frac{\partial o_i}{\partial w_k}$$

and

$$\lambda_i = \sum_{j:(i,j) \in I} \frac{-1}{1 + e^{o_i - o_j}} + \sum_{j:(j,i) \in I} \frac{1}{1 + e^{o_i - o_j}}$$

Here $o_i = f(\phi(x_i)) \in \mathbb{R}$ and $o_j = f(\phi(x_j)) \in \mathbb{R}$ are the outputs of the two branches of the model. Theorem 2 shows that instead of performing back-propagation for all possible pairs, we could first perform N forward passes to obtain $\{o_j\}$ and then calculate $\{\lambda_i\}$. Calculating $\{\lambda_i\}$ from $\{o_j\}$ in Equation 5.4 takes negligible time since this stage does not involve any neural network operation. Finally, we calculate a weighted sum of $\mathcal{O}(N)$ back-propagation and update the model parameters.

6 Experiment

Model Setup We fine tune the pre-trained BERT (Devlin et al., 2019) to learn the dialog feature extractor ϕ. We partition the 403 expert annotated dialog pairs into a 200-pair development set and a 203-pair test set. We set $K = 50$ for both the KNN label smoothing in Stage 2 and the KNN Shapley value calculation in Stage 3.

Model Details The details of extending BERT to encode multi-turn dialogs are as follows. Each dialog is represented as a sequence of tokens in the following input format: Starting with a special starting token $[CLS]$, we concatenate tokenized user and system utterances in chronological order with $[SEP]$ as the separators for adjacent utterance. In other words, we represent each dialog as a sequence: $[CLS]$, $S_{1,1}$, $S_{1,2}$, ..., $[SEP]$, $U_{1,1}$, $U_{1,2}$, ..., $[SEP]$, $S_{2,1}$, $S_{2,2}$, ..., $[SEP]$ where $S_{i,j}$ and $U_{i,j}$ are the j^{th} token of the system and user utterance in the i^{th} turn. Following BERT, we also add a learned embedding to every token indicating whether it comes from user utterances or system utterances.

Model Comparisons and Ablations We compare CMADE to its several ablations (Table 4) and evaluate the performance on the testing set, which is annotated by experts. We also report the kappa

No.	Model	Test Acc.	Kappa	
			κ	SE
(1)	BERT-Classification	0.581	0.161	0.049
(2)	BERT-Regression	0.640	0.280	0.048
(3)	BERT-Pairwise	0.730	0.459	0.044
(4)	BERT-Pairwise+Dev	0.749	0.499	0.043
(5)	Stage 2	0.755	0.509	0.043
(6)	Stage 2 + 3	0.764	0.529	0.042
(7)	Stage 3	0.714	0.429	0.045
(8)	Stage 1	0.620	0.241	0.048
(9)	Stage 1 + 3	0.788	0.628	0.039
(10)	Stage 1 + 2	0.837	0.673	0.037
(11)	**CMADE**	**0.892**	**0.787**	**0.031**

Table 4: Test accuracy and kappa agreement comparison among variants of CMADE.

agreement (Cohen, 1968) (kappa κ and Standard Error SE) between the predicted output and the expert annotations. (1) BERT-Classification and (2) BERT-Regression fine tune the pre-trained BERT to perform a 5-class classification and regression respectively directly using the noisy self-reported ratings. To test BERT-Classification on dialog pairs, we apply the DEX trick (Rothe et al., 2015) to get a floating-point number of predicted rating and thus get rid of the cases when the model predicts the dialog pairs as tie. (3) BERT-Pairwise shares the same model architecture with CMADE. It constructs dialog pairs for training by randomly sample dialog pairs x_i and x_j and derive $z_{i,j}$ by comparing the corresponding self-reported user rating y_i and y_j. We discard the pairs with equal y_i and y_j. (4) BERT-Pairwise+Dev augments (3) by adding the 200 expert annotated dialog pairs in the development into the training data. We also compare the variants of CMADE which skips one or two of the three stages.

Results Our first takeaway is that vanilla classification or regression formulation might not be the best way to formulate the problem of learning a dialog evaluation model. As shown in Table 4, pairwise architecture (BERT-Pairwise, 0.73) is better than classification (BERT-Classification, 0.53) or regression (BERT-Regression, 0.64) in this problem. Similar to our observation, the research community in computer vision has long observed that both vanilla classification and regression formulation has drawbacks in age estimation (Rothe et al., 2015; Niu et al., 2016; Zhang et al., 2017).

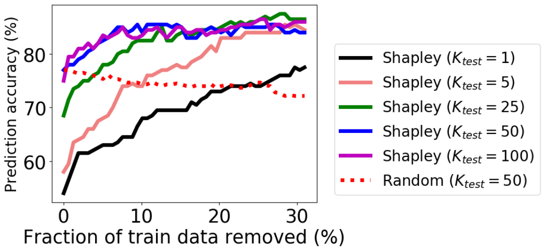

Figure 2: Removing training data with low Shapley value improves the performance of the KNN regressor.

Our second takeaway is that denoising algorithm that is more aggressive usually makes stronger assumptions on the quality of feature representations. Therefore, it helps to create a denoising pipeline that starts with better feature representation learning and less aggressive denoising algorithm to learn better feature representation before applying the more aggressive denoising algorithms. As shown in Table 4, our three-stage denoising pipeline CMADE (Acc. 0.892) significantly outperforms all baselines by a large margin. Although (8) Stage 1 does not directly provide high accuracy (Acc. 0.620), the feature representation it learned is extremely important. Without Stage 1, both (5) Stage 2 (Acc. 0.755) and (6) Stage 2 + Stage 3 (Acc. 0.763) perform worse.

Since the KNN label smoothing is performed on the feature space, we expect the smoothing performs worse without self-supervised dialog feature representation learning in Stage 1. However, they still work better than baseline (1) (2) (3) which are models that do not account for the noise in data. This is because we use the pre-trained BERT to initialize our dialog encoder ϕ and thus ϕ is still able to provide some useful features for Stage 2. In addition, we observe that denoising with data Shapley in Stage 3 requires better dialog feature representation. (7) Stage 3 (Acc. 0.714) performs even worse than BERT-Pairwise (0.730) without good representations to perform the Shapley denoising algorithm. Skipping Stage 2 also hurts performance (Acc. 0.788). However, it does not mean that Shapley denoising in Stage 3 is not powerful. We observe a large performance gain in applying stage 3 after stage 1 and stage 2 (Acc. 0.837 v.s. 0.892). Finally, we note that adding the expert annotated development set directly into the training data is much less efficient compared to using the development set for data Shapley to denoise. BERT-Pairwise+Dev only reaches an accuracy of 0.749.

6.1 Analysis

Additional analysis We also present an analysis to show how Shapley denoising works as shown in

Figure 2. We examine the Shapley value for each training datum in Stage 3. We first show an example dialog with a negative Shapley value in Table 5. According to the Shapley value, we remove data points one by one starting from the least valuable to the most valuable. Each time, after the point is removed, we create new KNN regressor models $K = 1, 5, 25, 50, 100$ on the remaining dialogs and labels and evaluate them on the test set with expert annotations. We extract the features of the remaining dialogs using the dialog encoder ϕ tuned by Stage 1 and Stage 2. As shown in Figure 2, removing training data with low Shapley values increases the performance to a certain point before convergence for K of all choices. We observe a similar trend when re-training a model on the remaining data. In contrast, removing data randomly decreases the performance on the test set. In addition, larger K has a better performance, which validates the denoising effect of KNN with large K.

7 Conclusion

The ultimate chatbot evaluation metric should be user-centric, as chatbots are there to provide human with an enjoyable experiences. Previously Likert-score based self-reported rating is the de-facto standard for current dialog evaluation . However, our analysis indicates that self-reported dialog ratings are skewed (J-shape), noisy and insensitive due to bias and variance among different users. We propose a three-stage denoising pipeline CMADE to reduce self-reported ratings and, at the same time, build an automatic comparison-based automatic dialog quality predictor. CMADE's results highly correlate with expert judgments on pair-wise dialog comparison ratings (89.2% agreement, 0.787 Kappa).

User: the whole harry potter series
Sys: whoops i think i misheard you. can you repeat the movie name please?
User: all the harry potter series
Sys: ah, i don't know any movies by that name. sorry if i sometimes confuse you saying something else with a movie name ...

...

Sys: oh, okay. anyways, i really enjoyed watching black panther and wonder woman recently. what's your favorite superhero movie?
User: super man
Sys: what was that? can you please say the movie name again?
User: stop i gotta go

Table 5: An example dialog with noisy self-reported rating (Score-5) identified by data shapley.

Acknowledgments

We would like to sincerely thank ACL 2020 Chairs and Reviewers for their review efforts and helpful feedback. We thank Yu Li for his insightful guidance and support in shaping this project. We thank Boxin Wang for helpful discussions on data Shapley. We would also like to extend our gratitude to Yanbang Wang, Youzhi Tian, Weiyan Shi and Michihiro Yasunaga for their valuable feedback and suggestions.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*, pages 65–72. Association for Computational Linguistics.

Dan Bohus and Eric Horvitz. 2009. Learning to predict engagement with a spoken dialog system in open-world settings. In *SIGDIAL Conference*, pages 244–252. The Association for Computer Linguistics.

Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. 2006. Learning to rank with nonsmooth cost functions. In *NIPS*, pages 193–200. MIT Press.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. 2005. Learning to rank using gradient descent. In *ICML*, volume 119 of *ACM International Conference Proceeding Series*, pages 89–96. ACM.

Chun-Yen Chen, Dian Yu, Weiming Wen, Yi Mang Yang, Jiaping Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhowmick, Shreenath Iyer, et al. 2018. Gunrock: Building a human-like social bot by leveraging large scale real user data. *Alexa Prize Proceedings*.

Jan Chorowski and Navdeep Jaitly. 2017. Towards better decoding and language model integration in sequence to sequence models. In *INTERSPEECH*, pages 523–527. ISCA.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Shay B Cohen, Eytan Ruppin, and Gideon Dror. 2005. Feature selection based on the shapley value. In *IJCAI*, volume 5, pages 665–670.

Peter J Danaher and Vanessa Haddrell. 1996. A comparison of question scales used for measuring customer satisfaction. *International Journal of Service Industry Management*, 7(4):4–26.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Pradeep Dubey. 1975. On the uniqueness of the shapley value. *International Journal of Game Theory*, 4(3):131–139.

Robert East, Kathy Hammond, and Malcolm Wright. 2007. The relative incidence of positive and negative word of mouth: A multi-category study. *International journal of research in marketing*, 24(2):175–184.

Maxine Eskénazi, Shikib Mehri, Evgeniia Razumovskaia, and Tiancheng Zhao. 2019. Beyond turing: Intelligent agents centered on the user. *CoRR*, abs/1901.06613.

Hao Fang, Hao Cheng, Maarten Sap, Elizabeth Clark, Ari Holtzman, Yejin Choi, Noah A. Smith, and Mari Ostendorf. 2018. Sounding board: A user-centric and content-driven social chatbot. In *NAACL-HLT (Demonstrations)*, pages 96–100. Association for Computational Linguistics.

Zunlei Feng, Weixin Liang, Daocheng Tao, Li Sun, Anxiang Zeng, and Mingli Song. 2019. Cu-net: Component unmixing network for textile fiber identification. *Int. J. Comput. Vis.*, 127(10):1443–1454.

Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. 2014. Bootstrapping dialog systems with word embeddings. In *Nips, modern machine learning and natural language processing workshop*, volume 2.

Johannes Fürnkranz and Eyke Hüllermeier. 2010. *Preference learning*. Springer.

Amirata Ghorbani and James Y. Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 2242–2251. PMLR.

David Godes and José C Silva. 2012. Sequential and temporal dynamics of online opinion. *Marketing Science*, 31(3):448–473.

Faruk Gul. 1989. Bargaining foundations of shapley value. *Econometrica: Journal of the Econometric Society*, pages 81–95.

Fenfei Guo, Angeliki Metallinou, Chandra Khatri, Anirudh Raju, Anu Venkatesh, and Ashwin Ram. 2018. Topic-based evaluation for conversational bots. *CoRR*, abs/1801.03622.

Prakhar Gupta, Shikib Mehri, Tiancheng Zhao, Amy Pavel, Maxine Eskénazi, and Jeffrey P. Bigham. 2019. Investigating evaluation of open-domain dialogue systems with human generated multiple references. *CoRR*, abs/1907.10568.

Nan Hu, Paul A Pavlou, and Jie Jennifer Zhang. 2017. On self-selection biases in online product reviews. *MIS Quarterly*, 41(2):449–471.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. 2019. Music transformer: Generating music with long-term structure. In *ICLR (Poster)*. OpenReview.net.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gürel, Bo Li, Ce Zhang, Costas J. Spanos, and Dawn Song. 2019a. Efficient task-specific data valuation for nearest neighbor algorithms. *PVLDB*, 12(11):1610–1623.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. 2019b. Towards efficient data valuation based on the shapley value. In *AISTATS*, volume 89 of *Proceedings of Machine Learning Research*, pages 1167–1176. PMLR.

Ilya Kulikov, Alexander H. Miller, Kyunghyun Cho, and Jason Weston. 2018. Importance of a search strategy in neural dialogue modelling. *CoRR*, abs/1811.00907.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. 2016. A persona-based neural conversation model. In *ACL (1)*. The Association for Computer Linguistics.

Margaret Li, Jason Weston, and Stephen Roller. 2019a. ACUTE-EVAL: improved dialogue evaluation with optimized questions and multi-turn comparisons. *CoRR*, abs/1909.03087.

Yu Li, Kun Qian, Weiyan Shi, and Zhou Yu. 2019b. End-to-end trainable non-collaborative dialog system. *arXiv preprint arXiv:1911.10742*.

Weixin Liang, Kai Bu, Ke Li, Jinhong Li, and Arya Tavakoli. 2018. Memcloak: Practical access obfuscation for untrusted memory. In *ACSAC*, pages 187–197. ACM.

Weixin Liang, Youzhi Tian, Chengcai Chen, and Zhou Yu. 2019. MOSS: end-to-end dialog system framework with modular supervision. *CoRR*, abs/1909.05528.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael 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 *EMNLP*, pages 2122–2132. The Association for Computational Linguistics.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *ACL (1)*, pages 1116–1126. Association for Computational Linguistics.

Vikram Sharma Mailthody, Zaid Qureshi, Weixin Liang, Ziyan Feng, Simon Garcia De Gonzalo, Youjie Li, Hubertus Franke, Jinjun Xiong, Jian Huang, and Wen-Mei Hwu. 2019. Deepstore: In-storage acceleration for intelligent queries. In *MICRO*, pages 224–238. ACM.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? *CoRR*, abs/1906.02629.

Allen Nie, Arturo L. Pineda, Matt W. Wright Hannah Wand, Bryan Wulf, Helio A. Costa, Ronak Y. Patel, Carlos D. Bustamante, and James Zou. 2019. Litgen: Genetic literature recommendation guided by human explanations. *CoRR*, abs/1909.10699.

Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. 2016. Ordinal regression with multiple output CNN for age estimation. In *CVPR*, pages 4920–4928. IEEE Computer Society.

Jekaterina Novikova, Ondrej Dusek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *EMNLP*, pages 2241–2252. Association for Computational Linguistics.

Ioannis Papaioannou, Amanda Cercas Curry, Jose L Part, Igor Shalyminov, Xinnuo Xu, Yanchao Yu, Ondrej Dušek, Verena Rieser, and Oliver Lemon. 2017. Alana: Social dialogue using an ensemble model and a ranker trained on user feedback. *Alexa Prize Proceedings*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. ACL.

Verónica Pérez-Rosas, Xinyi Wu, Kenneth Resnicow, and Rada Mihalcea. 2019. What makes a good counselor? learning to distinguish between high-quality and low-quality counseling conversations. In *ACL (1)*, pages 926–935. Association for Computational Linguistics.

Robert A Peterson and William R Wilson. 1992. Measuring customer satisfaction: fact and artifact. *Journal of the academy of marketing science*, 20(1):61.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. 2018a. Conversational AI: the science behind the alexa prize. *CoRR*, abs/1801.03604.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. 2018b. Conversational AI: the science behind the alexa prize. *CoRR*, abs/1801.03604.

Rasmus Rothe, Radu Timofte, and Luc Van Gool. 2015. DEX: deep expectation of apparent age from a single image. In *ICCV Workshops*, pages 252–257. IEEE Computer Society.

Vasile Rus and Mihai C. Lintean. 2012. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *BEA@NAACL-HLT*, pages 157–162. The Association for Computer Linguistics.

Verena Schoenmüller, Oded Netzer, and Florian Stahl. 2018. The extreme distribution of online reviews: Prevalence, drivers and implications. *Columbia Business School Research Paper*.

Edwin D. Simpson and Iryna Gurevych. 2018. Finding convincing arguments using scalable bayesian preference learning. *Trans. Assoc. Comput. Linguistics*, 6:357–371.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826. IEEE Computer Society.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. RUBER: an unsupervised method for automatic evaluation of open-domain dialog systems. In *AAAI*, pages 722–729. AAAI Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Anu Venkatesh, Chandra Khatri, Ashwin Ram, Fenfei Guo, Raefer Gabriel, Ashish Nagar, Rohit Prasad, Ming Cheng, Behnam Hedayatnia, Angeliki Metallinou, Rahul Goel, Shaohua Yang, and Anirudh Raju. 2018. On evaluating and comparing conversational agents. *CoRR*, abs/1801.03625.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. *CoRR*, abs/1506.05869.

Catherine Wah, Grant Van Horn, Steve Branson, Subhransu Maji, Pietro Perona, and Serge Belongie. 2014. Similarity comparisons for interactive fine-grained categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–866.

Sanghyun Yi, Rahul Goel, Chandra Khatri, Tagyoung Chung, Behnam Hedayatnia, Anu Venkatesh, Raefer

Gabriel, and Dilek Hakkani-Tür. 2019. Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. *CoRR*, abs/1904.13015.

Gal Yona, Amirata Ghorbani, and James Zou. 2019. Who's responsible? jointly quantifying the contribution of the learning algorithm and training data. *arXiv preprint arXiv:1910.04214*.

Dian Yu, Michelle Cohn, Yi Mang Yang, Chun-Yen Chen, Weiming Wen, Jiaping Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhowmick, Shreenath Iyer, Giritheja Sreenivasulu, Sam Davidson, Ashwin Bhandare, and Zhou Yu. 2019. Gunrock: A social bot for complex and engaging long conversations. In *EMNLP/IJCNLP (3)*, pages 79–84. Association for Computational Linguistics.

Georgios Zervas, Davide Proserpio, and John Byers. 2015. A first look at online reputation on airbnb, where every stay is above average. *Where Every Stay is Above Average (January 28, 2015)*.

Yunxuan Zhang, Li Liu, Cheng Li, and Chen Change Loy. 2017. Quantifying facial age by posterior of age comparisons. In *BMVC*. BMVA Press.