A Human-machine Collaborative Framework for Evaluating Malevolence in Dialogues

Yangjun Zhang
University of Amsterdam
y.zhang6@uva.nl

Pengjie Ren*
Shandong University
renpengjie@sdu.edu.cn

Maarten de Rijke
University of Amsterdam
& Ahold Delhaize
m.derijke@uva.nl

Abstract

Conversational dialogue systems (CDSs) are hard to evaluate due to the complexity of natural language. Automatic evaluation of dialogues often shows insufficient correlation with human judgements. Human evaluation is reliable but labor-intensive. We introduce a human-machine collaborative framework, HMCEval, that can guarantee reliability of the evaluation outcomes with reduced human effort. HMCEval casts dialogue evaluation as a sample assignment problem, where we need to decide to assign a sample to a human or a machine for evaluation. HMCEval includes a model confidence estimation module to estimate the confidence of the predicted sample assignment, and a human effort estimation module to estimate the human effort should the sample be assigned to human evaluation, as well as a sample assignment execution module that finds the optimum assignment solution based on the estimated confidence and effort. We assess the performance of HMCEval on the task of evaluating malevolence in dialogues. The experimental results show that HMCEval achieves around 99% evaluation accuracy with half of the human effort spared, showing that HMCEval provides reliable evaluation outcomes while reducing human effort by a large amount.

1 Introduction

Conversational dialogue systems (CDSs) are often trained to generate responses given unstructured, open-domain dialogues. Evaluation of CDS responses has drawn broad attention due to its crucial rule for CDS development (Deriu et al., 2020). Broadly speaking, there are two approaches to perform dialogue evaluation: *automatic* evaluation and *human* judgements (Finch and Choi, 2020). Automatic evaluation metrics such as appropriateness (Lowe et al., 2017), engagement (Zhang et al., 2020), are efficient but have low agreement with human judgements due to the diversity of responses (Liu et al., 2016), especially for word-overlap based metrics, such as BLEU (Papineni et al., 2002) and ROUGE (Lin and Hovy, 2002). More recently, training based methods, e.g., ADEM (Lowe et al., 2017), RUBER (Tao et al., 2018) and contextualized methods, e.g. BERT-based RUBER (Ghazarian et al., 2019), have been shown to have better agreement with human judgements. However, these methods are still not reliable enough: the Pearson correlation with human judgments is 0.44 for appropriateness (Lowe et al., 2017) and 0.55 for relevance (Ghazarian et al., 2019). To guarantee reliability of evaluation outcomes, our current best practice is to use human judgements. In terms of most evaluation aspects, e.g., appropriateness (Young et al., 2018), coherence (Ram et al., 2018) and empathy (Rashkin et al., 2019), human judgements simply show the highest reliability. Obviously, human judgments are more labor-intensive than automatic evaluation (Deriu et al., 2020).

The flaws of automatic evaluation and the lack of speed and scalability of human evaluation limits the speed at which the community can develop more intelligent CDSs. For example, as part of the daily research and development cycle of CDSs, we need to change the model design and retrain the model multiple times, on a daily or even hourly basis. Even if there is a minor change, we need to verify its performance again each time. For another example, CDS leaderboards are very popular recently as a means to provide platforms for fair comparison (Hou et al., 2019). There are usually dozens of models to evaluate, and new models are introduced everyday. Practical scenarios like the above two call for dialogue evaluation methods that are both reliable and efficient.

In this paper, we propose the *human-machine*

* Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5612–5623
August 1–6, 2021.

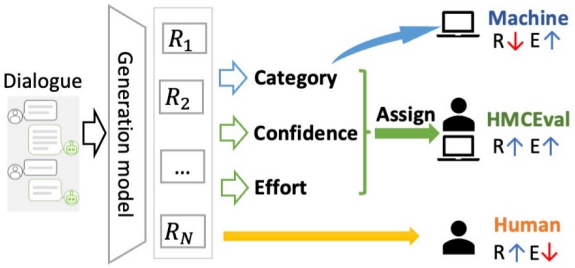


Figure 1: Human-machine collaborative evaluation (HMCEval) framework. $R_1, \ldots, R_N$ are the generated response samples to be evaluated. R and E are reliability and efficiency, respectively.

collaborative evaluation (HMCEval) framework for dialogue evaluation with the aim of balancing reliability and efficiency. HMCEval formulates the dialogue evaluation task as a sample assignment problem, i.e., if the machine can provide accurate outcomes, most evaluation samples should be assigned to the machine; otherwise, we should assign more samples to human evaluators. As shown in Figure 1, automatic evaluation has low reliability although the efficiency is high; human judgement has high reliability but it is labor-intensive; HMCEval beats the previous two methods in balancing reliability and efficiency. Finding a good balance between reliability and efficiency is non-trivial as the two desiderata are often in conflict with each other. Usually, reliability is improved at the expense of efficiency (Chaganty et al., 2018).

There are three main modules in *human-machine collaborative evaluation* (HMCEval), namely the *model confidence estimation* (MCE) module, the *human effort estimation* (HEE) module, and the *sample assignment execution* (SAE) module. First, the MCE module measures the confidence of predicted evaluation for each dialogue response based sample. Our implementation of MCE is based on three estimation methods, namely, BERT based maximum class probability (MCP), trust score (TS) (Jiang et al., 2018), and true class probability (TCP) (Corbière et al., 2019). TS and TCP have originally been introduced for images; we add a BERT layer to expand it to dialogues. Second, the HEE module estimates the effort. Our implementation is based on annotation time cost prediction by dialogue-related and worker-related features. Third, the SAE module decides whether a dialogue response sample should be assigned to a human or a machine for evaluation by maximizing the confidence and minimizing the (human) effort. We implement the module by integer linear programming (ILP).

We demonstrate the effectiveness of HMCEval on dialogue malevolence evaluation (Zhang et al., 2021). The main reason we choose this particular task is that dialogue malevolence is highly related to social good (Xu et al., 2020; Shi et al., 2020), which is of vital importance for CDSs, but it is hard to evaluate because of the need of deep semantic understanding (Das et al., 2020). We carry out experiments on the recently introduced malevolent dialogue response detection and classifying (MDRDC) dataset (Zhang et al., 2021). Our results show that the proposed HMCEval framework significantly surpasses machine evaluation and human judgement in terms of balancing reliability and effort. HMCEval achieves around 99% evaluation accuracy (compared to human evaluation) with as much as half of the human effort saved. The results demonstrate that HMCEval can be used for reliable and efficient evaluation of CDSs since the accuracy is high and the effort is significantly reduced compared to fully human evaluation.

2 Related Work

2.1 Evaluation of CDSs

Automatic evaluation for CDSs includes untrained methods and learning based methods. Early untrained methods, such as perplexity (Chen et al., 1998), and quality metrics BLEU (Papineni et al., 2002) and ROUGE (Lin and Hovy, 2002) are widely used for CDS but the aspects they evaluate are limited. Recent work based on word embeddings cover more aspects, such as distinct-n for diversity (Li et al., 2016) or average word embedding similarity for coherence (Luo et al., 2018). Most untrained methods have low agreement with human judgements (Liu et al., 2016) because machine responses are highly diversified, although a few metrics have sufficient agreement with human, i.e., a Pearson correlation of 0.69 for coherence (Luo et al., 2018).

To address the problem of low agreement with human judgments, learning based methods have been developed (Novikova et al., 2017; Tao et al., 2018). Lowe et al. (2017) propose ADEM to evaluate the appropriateness of responses. Tao et al. (2018) propose RUBER, which shows better agreement with human judgments than ADEM. RUBER is designed for relevance and similarity by blending relevance between the generated response with human ground truth and context. Several methods utilize pretrained language models such as BERT for automatic evaluation. Ghazarian et al.

(2019) propose contextualized RUBER, which outperforms RUBER. Similarly, a predictive engagement metric is built by utilizing user engagement score (Ghazarian et al., 2020); quality is evaluated by transformer based language models without reference response (Nedelchev et al., 2020). The above methods cover more aspects and integrate linguistic features (Tao et al., 2018), thus the agreement with human judgement is higher than most word-overlap based methods. However, for most of the metrics, the model performance still has space to improve, for instance, the accuracy of engagement is 0.76 (Ghazarian et al., 2020). Our proposed HMCEval framework could be applied to these metrics and improve general evaluation reliability with an acceptable amount of human effort.

Human judgement is applied in common evaluation aspects including fluency, consistence, relevance, appropriateness, coherence, quality for CDSs (Finch and Choi, 2020). It is reliable, yet expensive and time intensive, especially for large scale evaluation (Hou et al., 2019). In order to guarantee reliability, agreement among different workers is needed, which makes the high effort problem more severe (Das et al., 2020).

Unlike the methods listed above, the HMCEval framework specifically aims to balance reliability and human effort for the evaluation of CDSs.

2.2 Human-machine collaboration

Human-machine collaboration hybridizes machine prediction and human judgements. Previous research mostly focuses on using human judgments to help label the low reliability samples (Callaghan et al., 2018; Kyono et al., 2018; Gates et al., 2020). Earlier research gives human the output of an automatic model and lets human decide whether the model prediction is reliable (Lasecki et al., 2012). However, people tend to ignore the predictions of a model if it makes mistakes (Dietvorst et al., 2015) since they are not tolerant to model mistakes. In such cases, predictive results are not fully utilized and human effort increases. At the same time, there is a possibility that human annotators mistakenly follow the outputs of a model with errors (Cummings, 2004). Both situations lead to failure of human-machine collaboration.

The core problem is to determine when a human annotator should trust a model. Confidence estimation for a model's prediction has been proposed to help improve overall accuracy, correctness etc. for human-machine collaboration. Callaghan et al. (2018) develop a hybrid cardiogram classification human-machine collaborative (HMC) framework, which achieves better performance than a classifier by itself and uses less expert resources compared to expert classification by itself. Kyono et al. (2018) develop a Man and Machine Mammography Oracle that improves overall breast cancer diagnostic accuracy, while reducing the number of radiologist readings. Gates et al. (2020) use Abstrackr based a HMC screening method to screen relevant title and abstract for paper reviews, which could save time of reviewers and have little risk of missing relevant records. However, the above methods select the top-k most unreliable samples and do not consider effort division between human and machine. Chaganty et al. (2018) are the first to combine machine and human evaluation to obtain a reliable estimate at lower cost than human alone on summarizing and open-source question answering, with cost reduction only 7–13%. Ravindranath et al. (2020) build a highly cost-efficient face recognition HMC framework that outperforms both a machine-based method and a fully manual method, with both reliability and effort considered. However, the methods introduced previously are not suitable for HMC evaluation for dialogue because of focusing on non-dialogue tasks, low cost reduction, or not considering both reliability and effort.

Our proposed framework is purpose-built for dialogue evaluation. It leverages both human judgement and machine prediction by assigning low confidence machine-generated samples to human workers, while minimizing overall human effort.

3 Methodology

3.1 Overview

Suppose we have a set of M samples $\{(C_i, \hat{x}_i)\}_{i=1}^{M}$ to be evaluated. Here, C_i is the dialogue context and $\hat{x}_i$ is a response generated by a CDS model $f_g(C) \rightarrow \hat{x}$. Below, we propose a method to achieve reliable and efficient evaluation of the M samples under the constraint that a human can annotate at most $N \ll M$ samples. We propose the *human-machine collaborative evaluation* (HMCEval) framework to solve this task. HMCEval is divided into three modules: sample assignment execution (SAE), model confidence estimation (MCE) and human effort estimation (HEE).

3.2 SAE module

The optimization problem of assigning M samples to a human or machine can be solved by tractable integer linear programming, which is NP-complete (Papadimitriou and Steiglitz, 1998). First, we introduce the decision variable z_i to denote the sample assignment to a human or machine:

$$z_i = \begin{cases} 0, & \text{sample } i \text{ is assigned to a human;} \\ 1, & \text{sample } i \text{ is assigned to machine.} \end{cases} \quad (1)$$

Second, we define two ILP objectives that try to maximize the overall confidence and minimize the overall effort, respectively:

$$\begin{aligned} &\max \sum_{i=1}^{M} \hat{a}_i z_i + \sum_{i=1}^{M} b_i (1 - z_i), \\ &\min \sum_{i=1}^{M} k_i z_i + \sum_{i=1}^{M} \hat{l}_i (1 - z_i), \end{aligned} \quad (2)$$

where (a) M is the total number of samples to evaluate generated by the generation model $f_g(C) \rightarrow \hat{x}$; (b) $\hat{a}_i \in [0, 1]$ is the model confidence for evaluating sample i; (c) b_i is the human confidence for evaluating sample i; (d) k_i is the machine effort for evaluating sample i; and (e) $\hat{l}_i \in [0, 1]$ is the human effort for evaluating sample i.

We use the weighted sum method (Marler and Arora, 2010) to solve Eq. 2 so as to get the optimal z_i. The objective function in Eq. 2 is transformed into:

$$\begin{aligned} \max \Bigg[&\sum_{i=1}^{M} \hat{a}_i z_i + \sum_{i=1}^{M} b_i (1 - z_i) - \\ &\lambda \left(\sum_{i=1}^{M} k_i z_i + \sum_{i=1}^{M} \hat{l}_i (1 - z_i) \right) \Bigg], \end{aligned} \quad (3)$$

subject to

$$\begin{aligned} &\sum_{i=1}^{M} z_i \geq M - N \\ &b_i = 1 \text{ for } i = 1, \ldots, M \\ &k_i = 0 \text{ for } i = 1, \ldots, M \\ &\lambda \geq 0. \end{aligned} \quad (4)$$

The constraints are motivated as follows: (a) the number of samples assigned to a human is less than or equal to N; (b) human confidence is assumed to be 1; (c) machine effort is assumed to be 0; and (d) λ is greater than 0. N and λ are two parameters that we use to balance reliability and effort; λ is a trade-off parameter that controls the contribution of two objectives to the overall objective, as shown in Eq. 3; and N controls the total samples assigned to a human. As N gets larger or λ gets smaller, the overall evaluation is more reliable but needs more human effort. As N gets smaller or λ gets larger, the overall evaluation costs less human effort but gets less reliability.

3.3 MCE module

Given a machine evaluation model (usually a classification model (De Mattei et al., 2020)) $f_c(C, \hat{x}) \rightarrow \hat{y}$, where $\hat{y}$ is the evaluation result (usually a category, e.g., malevolence or non-malevolence), the MCE module aims to recognize how confident the evaluation $\hat{y}$ is. In this work, we investigate three confidence estimation methods, namely maximum class probability (MCP), trust score (TS) and true class probability (TCP).

MCP is a basic method that directly uses the classification probabilities to measure the confidence. Based on the dataset $\{(C_j^{'}, x_j), y_j\}_{j=1}^{Q}$, we build a BERT-based classifier as a machine evaluation model f_c. MCP is the softmax probability of the evaluation result $\hat{y}$. Formally, $\mathrm{MCP}(C^{'}, x) = P(Y = \hat{y} | w, C^{'}, x)$.

TS is a confidence measurement that estimates whether the predicted category of a test sample by a classifier can be trusted. It is calculated as the ratio between the Hausdorff distance from the sample to the non-predicted and the predicted categories (Jiang et al., 2018). First, the training data is processed to find k-NN radius based α-high-density-set $\hat{H}(\tilde{C}^{'}_{train}, \tilde{x}_{train})$, where $\{\tilde{C}^{'}_{train}, \tilde{x}_{train}\}$ is the output of feeding training samples $\{(C^{'}_{train}, x_{train})\}$ into the BERT layer of f_c. This part is different from the original TS work designed for images (Yu et al., 2019). Then, for a given test sample, we predict the ratio of distances, which is the TS value. Formally, $\hat{a} = d(C_j^{'}, x_j, \hat{H}_1) / d(C_j^{'}, x_j, \hat{H}_2)$, where $\hat{H}_1$ is the high density set of the non-predicted category, $\hat{H}_2$ is the high density set of the predicted category. The estimated TS is normalized within 0 and 1 by min-max normalization.

As for TCP, the estimation is obtained by a learning-based method. Similar to TS, the original confidence network for TCP estimation is also built for images (Corbière et al., 2019). We expand it into a BERT-based confidence network for CDSs. The TCP estimation part f_{conf}

is based on the BERT-classifier f_c. Formally, $f_{conf}(C, \hat{x}, f_c, f_g) \rightarrow \hat{a} \in [0, 1]$, where f_g is the generation model. We pass the features from the BERT layer of f_c and feed them into a confidence network implemented by a succession of dense layers with a sigmoid activation to get the confidence scalar.

We define an MSE loss to train TCP: $L_{conf} = \frac{1}{Q}\sum_{i=1}^{Q}(\hat{a}(C_i^{'}, x_i, \theta) - a^*(C_i^{'}, x_i, y_i^*))^2$, where $a^*(C_i^{'}, x_i, y_i^*)$ is the target confidence value. During inference, the ground truth TCP score is calculated based on the BERT-based classifier: $\text{TCP}(C^{'}, x, y^*) = P(Y = y^*|w, C^{'}, x)$, where y^* is the true category.

3.4 HEE module

The HEE module is designed for estimating the human effort $\hat{e}$. In this work, we use time cost, i.e., the time spent for each annotation, to represent human effort. We implement the time cost estimation model f_l with random forest regression (Liaw et al., 2002): $f_l(h(C, \hat{x})) \rightarrow \hat{l} \in [0, 1]$, h is the feature extraction function.

There are two groups of features, namely dialogue related features and worker related features; see Table 5. The dialogue related features are: (a) 'total turns': total number of turns in a dialogue; (b) 'malevolent turns': total number of malevolent turns in a dialogue; for prediction, we use the BERT-classifier results; (c) 'non-malevolent turns': total number of non-malevolent turns in a dialogue; for prediction, we use the BERT-classifier results. (d) 'first submission or not': if this is the first time the worker does this task, the value is 1, else 0; (e) 'paraphrased turns': some turns are paraphrased; we calculate the total number of such turns; (f) 'total length': total number of tokens in the dialogue; (g) 'FK score': the result of a readability test, based on (Kincaid et al., 1975); (h) 'DC score': the result of a readability test, based on (Dale and Chall, 1948); (i) 'contains malevolent turn or not': if the dialogue contains a malevolent turn, the value is 1, else 0; and (j) 'perplexity score': we use BERT as a language model to calculate the perplexity (Gamon et al., 2005). The worker related features are: (a) 'worker test score': this is based on a test designed to test workers' ability to annotate the dialogue according to the gold standard annotation (Zhang et al., 2021); and (b) 'approval rate ranking': we rank workers by their lifetime approval rate in ascending order, and use the index; lower approval rate workers (i.e., with a smaller index) usually spend less time on annotations.

To train the time cost estimation model f_l, we need the annotation time spent on each response. However, for each individual response, the time spent is relatively short; as a consequence, the influence of noise such as attention, click time, may be relatively large and make the data unreliable as training data. Therefore, we use the annotation time spent on each dialogue instead of each response as time cost target, and it is normalized within 0 and 1 using min-max normalization. For the SAE module and effort assessment, we use the average time per turn of each dialogue as the time cost $\hat{l}$ for each response. In addition, there are multiple human annotator submissions for inter-annotator agreement; we filter out the data points that disagree with the agreed annotation; then we choose the data point with a higher annotator test score; if the test scores are same, we randomly choose one.

4 Experimental Setup

4.1 Dataset

We carry out experiments on the MDRDC dataset which is initially built for malevolent dialogue detection and classification (Zhang et al., 2021). The dataset consists of 6,000 dialogues, with 21,081 non-malevolent utterances and 10,299 malevolent utterances. The dataset also includes MTurk information, e.g., the time spent on each annotation. We follow the original paper to split the dataset into train, validation and test with a ratio of 7:1:2.

4.2 Implementation details

In terms of the responses by the generation model f_g, in our implementation, we use the original responses by a human for evaluation. The MCE module is implemented by a BERT-based classifier and a BERT-based confidence network. First, for the BERT-based classifier, we add a softmax layer on top of the '[CLS]' token. It is fine-tuned with 4 epochs since it is already pretrained on a large dataset. The vocabulary size is 30,522. Dialogue context and the current response are concatenated with the '[SEP]' delimiter. We consider the previous three dialogue utterances (if any) as context. We set the max sequence length to 128, the batch size to 64, the dropout ratio to 0.1, and the learning rate is 5e-5. Second, the BERT-based confidence network is attached to a BERT-classifier. It is composed of 5 dense layers, following previous work (Corbière et al., 2019). As for max sequence

length, batch size, dropout ratio, and learning rate, these are the same as for the classifier. The confidence network is trained with a maximum of 30 epochs, with early stopping if the validation loss does not improve for 10 epochs. The HEE module is implemented by a random forest regression model; the max number of estimators in this study is 100; only the features related to time cost are selected for annotation time cost prediction, with a maximum feature size of 10. We use the MIP package to implement ILP for the SAE module[1] with the Coin-or branch-and-cut solver (Mitchell, 2002). The search stops when it reaches a feasible solution. All the neural models are trained on GeForce GTX TitanX GPUs.

4.3 Metrics

We use reliability metrics and effort metrics to assess overall performance. The reliability metrics are precision, recall, F1-score, and accuracy. We calculate the macro score of precision, recall and F1 as the categories are imbalanced (Hossin and Sulaiman, 2015). The effort metrics include human ratio and time cost. Human ratio is the ratio of samples assigned to a human. Time cost is the total time required for a human to annotate the samples. We use AUC, and top-k accuracy to assess the different MCE implementations (Ouni et al., 2017). We rank the confidence in descending order and calculate the accuracy at top-50%. Top-50% accuracy measures how well the MCE predictions work for the top-50% most confident samples. We use mean square error (MSE), rooted mean square error (RMSE), mean absolute error (MAE) and R^2 to assess the HEE module. MSE, RMSE, MAE are calculated between the predicted time cost and real time cost. We also use the Pearson and Spearman correlation scores to analyze the correlation between features and real time cost.

5 Results and Analysis

5.1 Reliability and efficiency

To determine how HMCEval compares to human evaluation and machine evaluation in balancing reliability and efficiency, we report the results in Table 1. HMCEval outperforms both human and machine evaluation in balancing reliability and efficiency. More importantly, HMCEval, with half of the human effort spared, achieves reliability that is close to human reliability. First, compared to human evaluation, HMCEval arrives at 98.5% of human accuracy but the human effort decreases by 50.0%. This means that HMCEval is much more efficient than human evaluation, while the reliability is close to human. Second, compared to machine evaluation, the precision, recall, F1-score and accuracy of HMCEval increase by 20.2%, 21.5%, 21.0%, and 14.3%, respectively. This means that HMCEval has higher reliability than machine evaluation. In sum, therefore, HMCEval surpasses both human and machine evaluation in balancing reliability and efficiency.

Table 1: Reliability and efficiency of HMCEval w.r.t. human and machine evaluation ($N/M = 0.5$).

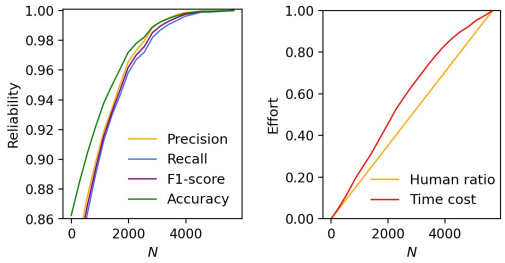
Metric	Machine	Human	HMCEval
Reliability			
Precision	0.818	1	0.983
Recall	0.803	1	0.976
F1-score	0.810	1	0.980
Accuracy	0.862	1	0.985
Efficiency			
Human ratio	0	1	0.500
Time cost	0	1	0.500

5.2 Influence of N and λ

To investigate how N and λ, two parameters for the SAE module that balance the reliability and effort, influence the performance of HMCEval, we first fix λ and vary N/M from 0 to 1 with a step size of 0.05, where M is the total number of samples to evaluate. Then, we fix N and vary λ from 0 to 45 with a step size of 0.1. The results are shown in Figure 2 and 3.

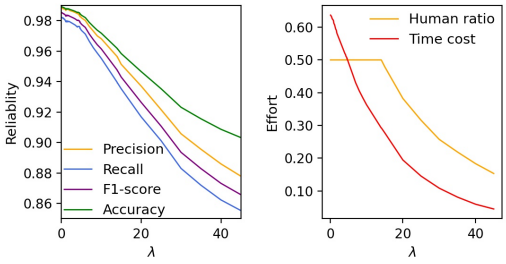
Influence of N. Generally, as N increases, HMCEval has better reliability, nevertheless the human effort increases. From Figure 2, we can see that when λ is fixed, as N gets larger, the precision, recall, F1-score and accuracy increase, but human ratio and time cost also increase. With larger N, more samples are assigned to a human, so the overall evaluation results are more reliable, but this requires a bigger human annotation effort. The marginal reliability benefit of assigning more samples to a human decreases as N gets larger. Figure 2(a) shows that as N increases, the reliability increases sharply at the beginning but the increase levels off when $N > 2,500$. The samples assigned to a human when $N < 2,500$ have lower model confidence, i.e., it is very likely that those samples are given inaccurate evaluation by machine. But when $N > 2,500$, samples with higher model confidence are also assigned to human which yields a

[1] https://python-mip.com



(a) Reliability. (b) Effort.

Figure 2: Influence of N with $\lambda = 0.1$.



(a) Reliability. (b) Effort.

Figure 3: Influence of λ with fixed N ($N/M = 0.5$).

limited return in terms of reliability.

Influence of λ. As λ increases, HMCEval gets more efficient, while the reliability gets worse. As shown in Figure 3, when λ increases, the human ratio stays at 0.5, and after a certain pivotal point, it decreases sharply. The time costs keep decreasing. The precision, recall, F1-score and accuracy decreases rapidly. With larger λ, the SAE objective puts a bigger emphasis on efficiency, so HMCEval gets more efficient but less reliable.

5.3 Module analysis

Analysis of the SAE module. By adjusting the λ values, the SAE module can degenerate into a greedy algorithm (Gates et al., 2020). Table 2 shows the results with the human ratio set to a fixed value of N/M, i.e., 0.5. When $\lambda = 0$, the HEE module has no effect, so it has the worst efficiency and the best reliability. When $\lambda \to \infty$, i.e., 500, the MCE module contributes little to the objective, so it has the best efficiency but the worst reliability.

Analysis of the MCE module. For the MCE module, we analyze the effect of alternative implementations. As shown in Figure 4, TS outperforms MCP and TCP. Specifically, when the human ratio is fixed to 0.5, TS achieves the best accuracy for different time costs. This means that TS has better model confidence estimation for the samples with higher confidence. As shown in Table 3, for the top-50% samples ranked by model confidence,

Table 2: Analysis of the SAE module.

Metric	MCE	MCE+HEE	HEE
Reliability			
Precision	0.989	0.983	0.881
Recall	0.982	0.976	0.858
F1-score	0.985	0.980	0.869
Accuracy	0.989	0.985	0.906
Efficiency			
Human ratio	0.500	0.500	0.500
Time cost	0.650	0.500	0.135

TS has the best accuracy. MCP has the best AUC score, which means for all the M samples, MCP is the best. But the top-50% samples have more influence on the SAE module.

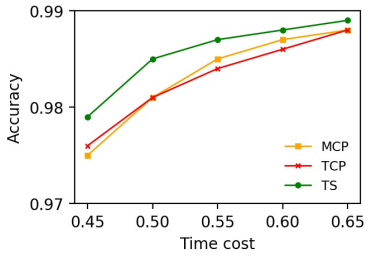


Figure 4: Performance of HMCEval with different MCE implementations ($N/M = 0.5$).

Table 3: Confidence prediction results comparison of MCE methods.

Metric	MCP	TCP	TS
AUC	**0.828**	0.823	0.825
Accuracy (top-50%)	0.977	0.975	**0.978**

Analysis of the HEE module. For the HEE module, we analyze the effect of different features. Adding worker related features helps to improve accuracy. As shown in Figure 5, SAE with both dialogue and worker related features has better accuracy than SAE with only dialogue related features when the human ratio is fixed to 0.5. Worker based features are useful for time cost estimation. This is confirmed by the results in Table 4. The results with both dialogue and worker related features are the best, with MSE, RMSE and MAE decreasing by 55.6%, 35.9%, 45.9%, and R^2 increasing by 76.2%. The HEE module is sufficient for time cost prediction since R^2 greater than 0.26 is sufficient for behavior related models (Cohen, 1988).

A correlation analysis between each feature and the real time cost is shown in Table 5. All the features, except perplexity, have significant Pearson or Spearman scores with the real time cost by workers. Most features show positive correlation. But two features, namely 'non-malevolent turns' and 'FK

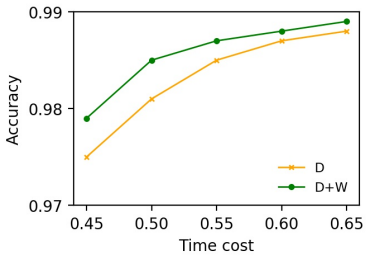


Figure 5: Feature analysis w.r.t. accuracy. (D: Dialogue related features, W: Worker related features.)

Table 4: Direct evaluation of the HEE module. (D: Dialogue related features, W: Worker related features.)

Metric	D	D+W
MSE	0.009	**0.004**
RMSE	0.092	**0.059**
MAE	0.061	**0.033**
R2	0.433	**0.763**

score' have a negative correlation with time cost: (a) non-malevolent responses are relatively easy to identify; and (b) a higher Flesch–Kincaid (FK) score means that the dialogue is easier to understand, which requires less time to annotate.

5.4 Performance at different turns

We analyze the effectiveness of HMCEval at different dialogue turns in Figure 6. As the dialogue evolves, HMCEval gets more reliable. It gets easier for the MCE module to detect malevolent responses with high confidence when more context information is available. The exception for turn seven and nine might due to the fact that the total number of utterances is small (less than 5% of the whole test set) and thus the results have high variance. The effort is not related to turn.

We also look into the 1.5% cases when HMCEval gives inaccurate evaluation, and some cases that require human judgement but are not assigned to a human. We find that these cases mostly have meaning extension, which means an extension of meaning of words with reference. For instance, 'I've commit 8 treasonous acts today and they still haven't put me in prison', this is actually a non-malevolent joke. However, the MCE module classified it to be malevolent with high confidence.

6 Conclusion and Future Work

In this work, we have introduced a human-machine collaborative evaluation framework (HMCEval) for reliable and efficient CDS evaluation. Experiments on the task of evaluating malevolence in dialogue responses show that HMCEval can achieve around 99% reliability with half human effort spared. A limitation of HMCEval is that given 50% samples

Table 5: Correlation analysis between time cost and different features for HMC module. ** and * indicate significance $p < 0.001$, $p < 0.05$, respectively.

Feature	Pearson	Spearman
Dialogue related features (D)		
Total turns	0.053^{**}	0.122^{**}
Malevolent turns	0.445^{**}	0.600^{**}
Non-malevolent turns	-0.236^{**}	-0.292^{**}
First Submission	0.342^{**}	0.263^{**}
Paraphrased turns	0.555^{**}	0.564^{**}
Total length	0.046^{**}	0.100^{**}
Readability (DC)	0.042^{*}	-0.001
Readability (FK)	-0.026^{*}	-0.053^{**}
Contains malevolent turn	0.432^{**}	0.603^{**}
BERT-perplexity	-0.008	0.001
Worker related features (W)		
Worker test score	0.162^{**}	0.049^{**}
Approval rate ranking	0.840^{**}	0.849^{**}

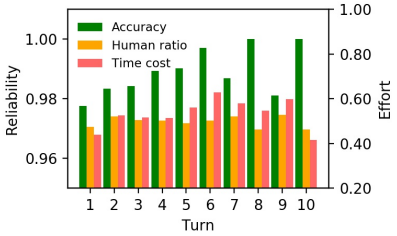


Figure 6: Accuracy and effort per turn with half human effort spared in average.

assigned to a human, 1.1–1.5% samples are evaluated inaccurately. This is due to contexts that consist of a small number of turns, or high confidence for some dialogues where language is used in a non-literal way. Although HMCEval could be generalized to several evaluation metrics of CDS, e.g., BERT-based RUBER and BERT-based engagement, for score-based metrics, suitable confidence estimation is required. In the future, we seek to improve the model confidence and human effort estimation by considering better neural architectures and more factors; we also plan to conduct a comprehensive and reliable analysis of the performance of current state-of-the-art CDS models by applying HMCEval to various evaluation aspects.

Acknowledgements

This research was funded by the China Scholarship Council and the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl.

Code

Our code is available at: https://github.com/repozhang/CaSE_HMCEval.

References

William Callaghan, Joslin Goh, Michael Mohareb, Andrew Lim, and Edith Law. 2018. Mechanicalheart: A human-machine framework for the classification of phonocardiograms. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):28:1–28:17.

Arun Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653.

Stanley F. Chen, Douglas Beeferman, and Roni Rosenfeld. 1998. Evaluation metrics for language models. In *DARPA Broadcast News Transcription and Understanding Workshop (BNTUW)*.

Jacob Cohen. 1988. *Statistical power analysis for the behavioral sciences (2nd edition)*. Hillsdale, L. Erlbaum Associates.

Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. 2019. Addressing failure prediction by learning model confidence. In *Advances in Neural Information Processing Systems*, pages 2902–2913.

Mary Cummings. 2004. Automation bias in intelligent time critical decision support systems. In *AIAA 1st Intelligent Systems Technical Conference*, page 6313.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Anubrata Das, Brandon Dang, and Matthew Lease. 2020. Fast, accurate, and healthier: Interactive blurring helps moderators reduce exposure to harmful content. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 8, pages 33–42.

Lorenzo De Mattei, Michele Cafagana, Felice Dell'Orletta, Malvina Nissim, and Albert Gatt. 2020. Changeit@evalita2020: Change headlines, adapt news, generate. In *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020), Online. CEUR.org*.

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2020. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, pages 1–56.

Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. 2015. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, 144(1):114.

Sarah E Finch and Jinho D Choi. 2020. Towards unified dialogue system evaluation: A comprehensive analysis of current evaluation protocols. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 236–245.

Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-level mt evaluation without reference translations: Beyond language modeling. In *Proceedings of the 10th EAMT Conference: Practical applications of machine translation*.

Allison Gates, Michelle Gates, Meghan Sebastianski, Samantha Guitard, Sarah A Elliott, and Lisa Hartling. 2020. The semi-automation of title and abstract screening: a retrospective exploration of ways to leverage abstrackr's relevance predictions in systematic and rapid reviews. *BMC Medical Research Methodology*, 20:1–9.

Sarik Ghazarian, Johnny Wei, Aram Galstyan, and Nanyun Peng. 2019. Better automatic evaluation of open-domain dialogue systems with contextualized embeddings. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 82–89.

Sarik Ghazarian, Ralph Weischedel, Aram Galstyan, and Nanyun Peng. 2020. Predictive engagement: An efficient metric for automatic evaluation of open-domain dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7789–7796.

Mohammad Hossin and MN Sulaiman. 2015. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. 2019. Identification of tasks, datasets, evaluation metrics, and numeric scores for scientific leaderboards construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5203–5213.

Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. 2018. To trust or not to trust a classifier. In *Advances in neural information processing systems*, pages 5541–5552.

J. Peter Kincaid, Robert P. Fishburne Jr, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Trent Kyono, Fiona J. Gilbert, and Mihaela van der Schaar. 2018. Mammo: A deep learning solution for facilitating radiologist-machine collaboration in breast cancer diagnosis. *arXiv preprint arXiv:1811.02661*.

Walter Lasecki, Christopher Miller, Adam Sadilek, Andrew Abumoussa, Donato Borrello, Raja Kushalnagar, and Jeffrey Bigham. 2012. Real-time captioning by groups of non-experts. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 23–34.

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

Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomforest. *R news*, 2(3):18–22.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*, pages 45–51.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.

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 *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126.

Liangchen Luo, Jingjing Xu, Junyang Lin, Qi Zeng, and Xu Sun. 2018. An auto-encoder matching model for learning utterance-level semantic dependency in dialogue generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 702–707.

R. Timothy Marler and Jasbir S. Arora. 2010. The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, 41(6):853–862.

John E Mitchell. 2002. Branch-and-cut algorithms for combinatorial optimization problems. *Handbook of applied optimization*, 1:65–77.

Rostislav Nedelchev, Jens Lehmann, and Ricardo Usbeck. 2020. Language model transformers as evaluators for open-domain dialogues. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6797–6808.

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

Ali Ouni, Raula Gaikovina Kula, Marouane Kessentini, Takashi Ishio, Daniel M German, and Katsuro Inoue. 2017. Search-based software library recommendation using multi-objective optimization. *Information and Software Technology*, 83:55–75.

Christos H. Papadimitriou and Kenneth Steiglitz. 1998. *Combinatorial Optimization: Algorithms and Complexity*. Courier Corporation.

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

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational AI: The science behind the Alexa prize. *arXiv preprint arXiv:1801.03604*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.

Saurabh Ravindranath, Rahul Baburaj, Vineeth N. Balasubramanian, NageswaraRao Namburu, Sujit Gujar, and C.V. Jawahar. 2020. Human-machine collaboration for face recognition. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pages 10–18.

Zheyuan Ryan Shi, Claire Wang, and Fei Fang. 2020. Artificial intelligence for social good: A survey. *arXiv preprint arXiv:2001.01818*.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *The 32nd AAAI Conference on Artificial Intelligence*.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots. *arXiv preprint arXiv:2010.07079*.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Kun Yu, Shlomo Berkovsky, Ronnie Taib, Jianlong Zhou, and Fang Chen. 2019. Do I trust my machine teammate? An investigation from perception to decision. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 460–468.

Yangjun Zhang, Pengjie Ren, and Maarten de Rijke. 2021. A taxonomy, dataset and benchmark for detecting and classifying malevolent dialogue responses. *Journal of the Association for Information Science and Technology*.

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. *ACL System Demonstrations*.

APPENDICES

We present additional details for reproducibility to the appendices. Specifically, we include corresponding validation performance for the main result (Appendix A), average runtime of each module and detailed information of parameters (Appendix B).

A Reliability and Efficiency of HMCEval for Validation

As for validation performance, we report the validation results of comparing HMCEval to machine evaluation and human evaluation in balancing reliability and efficiency, as shown in Table 6. HMCEval surpasses both human and machine evaluation in balancing reliability and efficiency for validation. On the one hand, compared to human evaluation, HMCEval achieves 98.2% of human accuracy with 50% human effort spared. This suggests that for the validation set, HMCEval is efficient than human evaluation, while the reliability is close to human evaluation. On the other hand, compared to machine evaluation, the precision, recall, F1-score and accuracy of HMCEval increase by 21.5%, 22.8%, 22.0% and 15.3%, respectively. Moreover, the results of the validation set and the test set are similar. Compared to results of the test set, reliability results of the validation set is slightly lower, but the difference is less than 0.5%, as shown in Table 1 (presented in Section 5) and Table 6.

Table 6: Reliability and efficiency of HMCEval w.r.t. human and machine evaluation for validation ($N/M = 0.5$).

Metric	Machine	Human	HMCEval
Reliability			
Precision	0.806	1	0.979
Recall	0.793	1	0.974
F1-score	0.800	1	0.976
Accuracy	0.852	1	0.982
Efficiency			
Human ratio	0	1	0.500
Time cost	0	1	0.500

B Runtime and Parameters

In terms of average runtime, we have three modules. The time costs for all the modules are acceptable. The MCE module has thee methods: MCP, TS and TCP. Their time costs are 0.5 hours, 0.1 hours, and 3.5 hours, respectively. The HEE module is implemented by random forest regression and the runtime is less than 10 minutes for 5-fold cross-validation. The SAE module is implemented by ILP and the runtime is around 2.5 hours.

In terms of parameters, the MCE module is neural network based. MCP and TS are estimated with the BERT-based classifier, which has 109.5 million parameters. TCP has an additional confidence network compared with MCP and TS. The confidence network part has 2.4 million parameters. The HEE module and the SAE module are not neural network based, we have included most of the information in the main manuscript. To add up, the SAE module is based on search. There are a total number of 10 thousand trials with different N and λ parameters. The best N and λ are chosen by reliability metrics and efficiency metrics. In Table 1 (presented in Section 5) and Table 6, we choose the final results with $\lambda = 4.6$ and $N = 0.5M$, where M is the number of the total samples to be evaluated.