RoBERTaIQ: An Efficient Framework for Automatic Interaction Quality Estimation of Dialogue Systems

Saurabh Gupta
gsaur@amazon.com
Alexa AI, Amazon
Seattle, USA

Xing Fan
fanxing@amazon.com
Alexa AI, Amazon
Seattle, USA

Derek Liu
derecliu@amazon.com
Alexa AI, Amazon
Seattle, USA

Benjamin Yao
benjamy@amazon.com
Alexa AI, Amazon
Seattle, USA

Yuan Ling
yualing@amazon.com
Alexa AI, Amazon
Seattle, USA

Kun Zhou
zhouku@amazon.com
Alexa AI, Amazon
Seattle, USA

Tuan-Hung Pham
hupha@amazon.com
Alexa AI, Amazon
Seattle, USA

Chenlei Guo
guochenl@amazon.com
Alexa AI, Amazon
Seattle, USA

ABSTRACT

Automatically evaluating large scale dialogue systems' response quality is a challenging task in dialogue research. Existing automated turn-level approaches train supervised models on *Interaction Quality* (IQ) labels or annotations provided by experts, which is costly and time-sensitive. Moreover, the small quantity of annotated data limits the trained model's ability to generalize to the long tail and out of domain cases. In this paper, we propose a learning framework that improves the model's generalizability by leveraging various unsupervised data sources available in large-scale conversational AI systems. We mainly rely on the following three techniques to improve the performance of dialogue evaluation models: First, we propose extending the RoBERTa model to encode multi-turn dialogues to capture the temporal differences between different turns. Second, we add two additional pretraining processes on top of enhanced multi-turn RoBERTa to take advantage of large quantity of existing historical dialogue data through self-supervised training. Third, we perform fine-tuning on IQ labels in a multi-task learning setup, leveraging domain-specific information from other tasks. We show that the above techniques significantly reduce annotated data requirements. We achieve the same F1 score on IQ prediction task as our baseline with only 5% of IQ training data and further beat the baseline by 5.4% absolute F1 score if we use all of the training data.

ACM Reference Format:
Saurabh Gupta, Xing Fan, Derek Liu, Benjamin Yao, Yuan Ling, Kun Zhou, Tuan-Hung Pham, and Chenlei Guo. 2021. RoBERTaIQ: An Efficient Framework for Automatic Interaction Quality Estimation of Dialogue Systems.

KDD '21, Aug 14–18, 2021, Virtual Event

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

In *Proceedings of DeMaL, 2nd International Workshop on Data-Efficient Machine Learning (KDD '21).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Large scale conversational agents like Amazon Alexa, Apple Siri, and Google Assistant have set a standard for conversational AI with the ability to integrate seamlessly across a wide range of functionalities. Such systems are complex in nature with many sequential components, such as Automatic Speech Recognition (ASR), Natural Language Understanding (NLU), Dialogue Manager, and Natural Language Generation. As the scope of these systems is increasing to cover more scenarios and applications, it becomes vital to automatically evaluate the response quality of these agents to estimate user satisfaction. In particular, identifying problematic responses where the user was left dissatisfied can be useful in improving dialogue agents over time with data driven learning [3, 16, 26].

Previous approaches for automated dialogue evaluation can be classified into *Dialogue-level* or *Turn-level*, based on whether we are evaluating multiple exchanges at once or each exchange (user's utterance and agent's response) individually. The PARADISE (PARAdigm for DIalogue System Evaluation) framework is the most well known evaluation framework proposed for evaluating dialogue-level user satisfaction [23]. In PARADISE, a linear regression model is fitted to predict the dialogue-level user satisfaction for a given set of manually extracted input features and user ratings. In contrast to rating the dialogue as a whole, approaches such as **Interaction Quality (IQ)** [19] were proposed to capture user satisfaction at turn level. Here, an SVM model [4] is learned based on the ratings provided by human annotators, while the input features were automatically extracted based on interaction parameters and emotions.

More recent approaches [1, 2, 11] extend the IQ framework and use models like Gradient Boosting Decision Tree (GBDT) [6], Recurrent Neural Networks (RNN) [17], and Long Short-Term Memory Networks (LSTM) [18] to encode the dialogue session. However, in addition to using textual data, these approaches also rely on

Table 1: Examples of Dialogue Sessions with different turn-level IQ labels (Unsatisfied = 1, Satisfied = 0)

#	Timestamp (in seconds)	Dialogue Session	IQ label
1	$\tau = 0$	[USER] Play hello [AGENT] Here's Hello, by Pop Smoke.	1
	$\tau = 4$	[USER] Stop [AGENT] <null>	
2	$\tau = 0$	[USER] Play maj and dragons. [AGENT] Sorry, I can't find that.	1
3	$\tau = 0$	[USER] Play hello [AGENT] Here's Hello, by Pop Smoke.	1
	$\tau = 6$	[USER] Play hello by Adele [AGENT] Here's hello by Adele	0
4	$\tau = 0$	[USER] Play hello [AGENT] Here's Hello, by Pop Smoke.	0
	$\tau = 60$	[USER] Play hello by Adele [AGENT] Here's hello by Adele	0
5	$\tau = 0$	[USER] show me shark videos [AGENT] Here's what I found *(playing video)*	1
	$\tau = 8$	[USER] play baby shark on amazon prime [AGENT] Here's Baby Shark , by Pinkfong , on Amazon Music.	0

input features generated by internal components, such as NLU/ASR confidence scores and Dialogue Status. These signals introduce dependencies on internal components and force the model to be system specific. As a result, our work does not leverage these signals; instead, we focus on more powerful model architectures that can capture user satisfaction using the textual and temporal information alone.

We hypothesize that user satisfaction can be inferred using explicit and implicit user/agent behaviors that exist in the dialogue session. Dialogues 1 and 2 in Table 1 are examples of explicit user and agent behavior, respectively. In dialogue 1, the user terminated the request as the agent did not play the right song. In dialogue 2, the agent failed to handle the request due to an error in entity resolution, caused by ASR error. Dialogues 3 and 4 capture user's intention implicitly and highlight the importance of temporal information. In dialogue 3, the user did not intend to listen to *Pop Smoke* and thus immediately interrupted the agent by rephrasing the original request. However, in dialogue 4, the user listened to "Hello, by Pop Smoke" for $\tau = 60$ seconds before issuing the next request. This arguably leads to the conclusion that the user intended to listen to *Pop Smoke* and *Adele* thereafter. Dialogue 5 shows why it is important to capture context from other turns. The agent's action in the first turn did not satisfy user's requirement as the user was looking for *Baby Shark* specifically.

These examples emphasize the significance of capturing dialogue context as precisely as possible, to correctly estimate user satisfaction. From the perspective of offline dialogue evaluation, the dialogue context should not only include the previous and the following turns, but also the temporal differences between different turns. To this end, we design a novel transformer based dialogue encoder, so as to utilize its self-attention mechanism [22] across tokens of different turns of the dialogue, while making the model aware of the temporal differences between turns. We build on top of RoBERTa encoder [13] and refer to our model as **RoBERTaIQ**.

Another major challenge imposed by automatic dialogue evaluation is collecting large amount of human annotations or IQ labels, which can be costly and time consuming. Recent advances in pretraining using self-attention encoder architectures like BERT [5] and RoBERTa [13] have been commonly used in many NLP applications. Such models are usually trained on massive general text corpora like English Wikipedia. However, the underlying difference of linguistic patterns between general text and dialogues makes existing pretrained language models less useful in practice. Wu et al. [24] have successfully shown that pretraining for task-oriented dialogues can be more useful than using general pretrained language models. However, there are only a few related works that leverage pretraining for automated dialogue evaluation. Liang et al. [10] learn dialogue feature representation with a self-supervised dialogue flow anomaly detection task, while Sinha et al. [20] train text encoders via noise contrastive estimation (NCE) [8]. Inspired by the success of domain-adaptive (DAPT) and task-adaptive pretraining (TAPT) [7, 9], we adopt the multi-stage pretraining process on large scale historical dialogue data and IQ task training data. Furthermore, we make our training process more data efficient by following the Multi-Task DNN learning framework for NLU [12]. We cast our learning process in a multi-task setting leveraging large amounts of cross-task data and regularization benefits. When fine-tuning, we learn to jointly predict turn-wise IQ label, domain and intent. The domain and intent signals are obtained from a separate NLU classification system and do not introduce additional annotation costs.

In summary, we make the following contributions:

- We design a novel transformer based dialogue encoder: RoBERTaIQ for inferring turn-level user satisfaction in multi-turn dialogues.
- We show the effectiveness of RoBERTaIQ in capturing dialogue context and temporal information across turns by comparing it with previous state of the art discourse-structure aware text encoders.
- We propose a data efficient learning framework to significantly reduce the amount of annotated data required for learning RoBERTaIQ. We leverage unlabelled historical dialogue data for pretraining. We perform fine-tuning in a multi-task learning setup to further utilize readily available signals like Domain and Intent. Unlike other works that use these signals as input features [1, 11], our model uses them as supervision signals to reduce training data (IQ labels) requirement.

The rest of the paper is organized as follows. Section 2 reviews existing work. Section 3 presents baselines and our approach for automatic dialogue evaluation. Section 4 presents our experimental results. Section 5 shows different ablation studies. We conclude our paper in Section 6. Appendices A and B contain hyperparameter information and case studies, respectively.

2 RELATED WORK

Recent works on evaluation of response quality in dialogue systems [1, 2, 11] are closely related to our work. While [2] use human engineered NLP features, [1, 11] propose IQ prediction models that use input features directly from raw dialogue turn contents and system

metadata (e.g. ASR/NLU scores). However, we see the reliance on system metadata as a limitation, and design our approach such that no system metadata is required as input features to the model. While the above approaches focus on dialogue evaluation in Spoken Langugage Understanding (SLU) systems, there is another line of work that focuses more on evaluation of open-domain chit-chat style dialogues. Lowe et al. [14] proposed a supervised approach called ADEM to mimic human annotator's assessment of response appropriateness, while Tao et al. [21] proposed an unsupervised method called RUBER. Both of these approaches use RNN based encoders. However, both ADEM and RUBER metrics result in poor correlation with human judgements [27]. Zhao et al. [27] propose RoBERTa-eval, which uses a powerful RoBERTa based text encoder to represent the dialogue context. Recently, Sinha et al. [20] propose MaUdE, which uses a BERT based text encoder to encode the utterances, followed by an RNN to model dialogue transitions. We adapt MaUdE and RoBERTa-eval to our use-case. We use them as baselines to analyze their shortcomings and design our dialogue encoder with enhanced contextual and temporal representations.

3 METHODOLOGY

In this section, we first define the notations and provide the problem definition. Next, we present the baseline model architectures adapted to our use-case: MaUdE and RoBERTa-eval. We then share the details of our proposed architecture: RoBERTaIQ that encodes a flattened dialogue text sequence and explain how each dialogue session is processed before inputting to this model. Next, we introduce how we obtain the datasets used for experiments, followed with explanation of the training procedure that involves pretraining and multi-task fine-tuning.

3.1 Notations and Problem Definition

We consider a dataset D of M multi-turn dialogue sessions, such that $D = \{S_j\}_{j=1}^{M}$, and every session S is an ordered set of N turns: $S = \{t_i\}_{i=1}^{N}$. Here i indicates the index of turn, and each turn t_i consists of a pair (Q_i, R_i), where Q_i is the user's query and R_i is the agent's response to query Q_i. Each turn t_i also has a timestamp τ_i associated with it, which is the time at which Q_i was received by the agent. Any two successive turns have a time gap of less than a minute. Given a dialogue session S and a **reference turn** $t_{ref} = t_i$ for some $i \in \{1, \ldots, N\}$, the goal of our model is to predict IQ_{score} of turn t_{ref}. $IQ_{score} = 0$ if the agent's response R_{ref} to query Q_{ref} is satisfactory from user's perspective, and 1 otherwise. We focus on offline turn-level dialogue evaluation, which means that we have both previous turns and following turns available at the time of evaluating t_{ref}.

3.2 Baseline models

3.2.1 MaUdE++. MaUdE (Metric for automatic Unreferenced dialogue Evaluation) was proposed by Sinha et al. [20] for online dialogue evaluation. Here, we adapt MaUdE's Dialogue-structure aware encoder for offline evaluation, and slightly modify the architecture such that it can encode other meta information about each turn, such as domain, intent, timestamp, active screen availability etc. We use similar metadata features as used by Ling et al. [11]. We refer to this modification as MaUdE++.

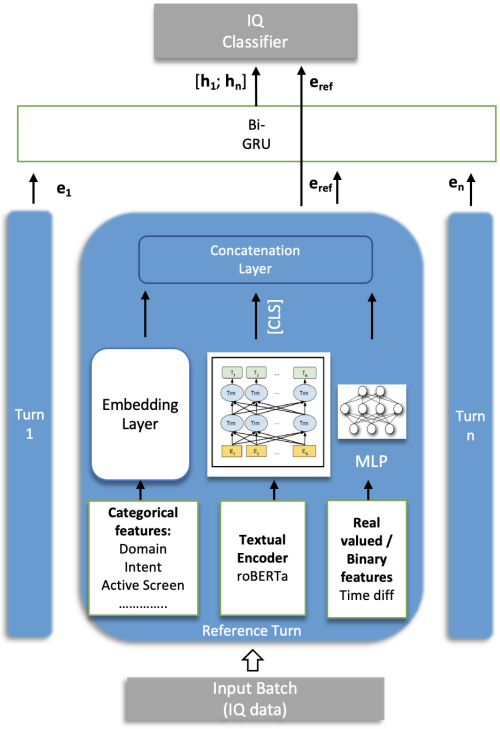

Figure 1: MaUdE++ (baseline model) architecture

As shown in Figure 1, MaUdE++ first computes the encoding for each turn and then passes the turn encodings through a bidirectional GRU to compute the dialogue session embedding. This session embedding is concatenated with other features and fed to the classifier for IQ prediction. Considering a dialogue session S with n turns: $\{(Q_1, R_1), ..., (Q_n, R_n)\}$, we compute the IQ score as:

$$\mathbf{f_i} = RoBERTa_{CLS}([CLS]; Q_i; [SEP]; R_i) \quad (1)$$

$$\mathbf{e_i} = (\mathbf{f_i}; \mathbf{meta_i}) \quad (2)$$

$$\overrightarrow{\mathbf{h_n}}, \overleftarrow{\mathbf{h_1}} = BiGRU(\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_n}) \quad (3)$$

$$IQ_{score} = \sigma(W.(\mathbf{e_{ref}}; \overrightarrow{\mathbf{h_n}}; \overleftarrow{\mathbf{h_1}})) \quad (4)$$

Here, [CLS] refers to a special token prefixed to user query, [SEP] refers to a special token inserted between the query and response, ";" denotes the concatenation operator, $\mathbf{e_i}$ refers to the encoding of turn t_i, $\mathbf{meta_i}$ refers to the concatenated encodings of categorical and real-valued features of turn t_i. $\overrightarrow{\mathbf{h_n}}$ and $\overleftarrow{\mathbf{h_1}}$ are the final hidden states of the bidirectional GRU in either direction and $\mathbf{e_{ref}}$ refers to the encoding of the *reference turn* for which we want to make the IQ score prediction. To compute the text representation $\mathbf{f_i}$, we use the [CLS] token encoding from the RoBERTa encoder. We start with a pretrained RoBERTa model and finetune it end-to-end with gradients coming from IQ classification loss.

3.2.2 RoBERTa-eval. RoBERTa-eval was proposed by Zhao et al. [27] as a robust dialogue response evaluator. It produces a vector $\mathbf{d}$ given a context c and a response R_{ref} and then finally calculates

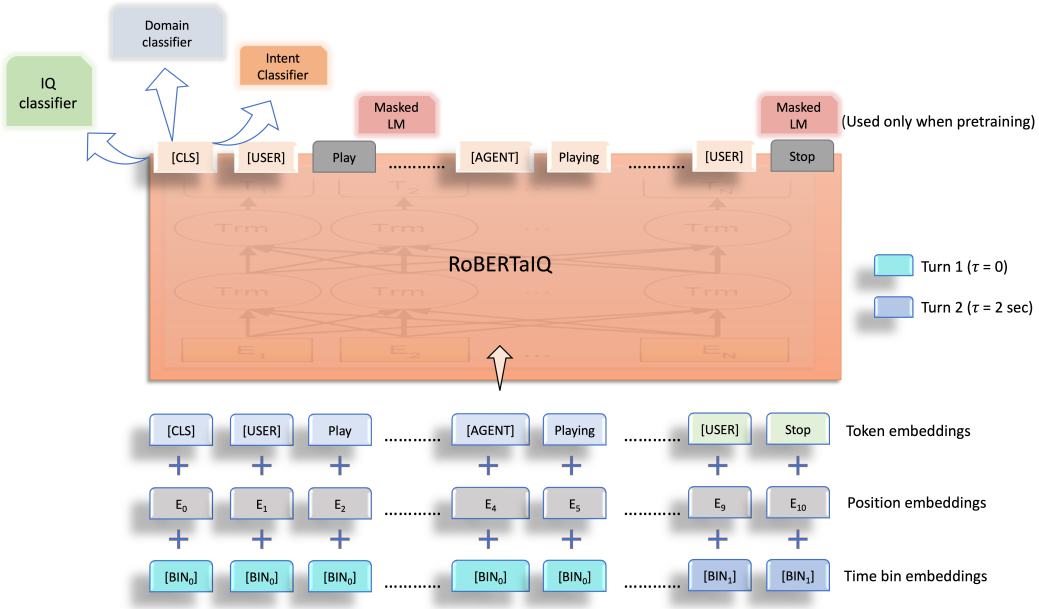

Figure 2: RoBERTaIQ model architecture

its score via a Multilayer Perceptron with a sigmoid function. Considering a dialogue session S with n turns: $\{(Q_1, R_1), ..., (Q_n, R_n)\}$:

$$c = [[CLS]; Q_1; [SEP]; R_1; [SEP]; Q_2; [SEP]; \dots Q_{ref}] \quad (5)$$

$$\mathbf{d} = RoBERTa_{CLS}(c; [SEP]; R_{ref}) \quad (6)$$

$$IQ_{score} = \sigma(MLP(\mathbf{d})) \quad (7)$$

Here, c, the dialogue context, is a flattened sequence of user queries and agent responses from previous turns, including the query of reference turn, for which we want to predict the IQ score. Note that a limitation of this model is that it can only encode the left dialogue context, i.e. turns that happened before Q_{ref}.

3.3 Our approach: RoBERTaIQ

Figure 2 shows the RoBERTaIQ model architecture diagram. Unlike previous works that rely on textual features and many other system specific signals [11, 15] which require feature engineering efforts, RoBERTaIQ relies solely on the textual features, i.e. user's utterances and agent's responses. RoBERTaIQ model is built on top of RoBERTa-base model with modifications at the input layer. To differentiate between the utterances and responses, we add two special tokens to the vocabulary, [USER] and [AGENT] and prefix them to each utterance and response respectively. By doing this, we create a single flat sequence for the whole dialogue. We limit the dialogue length to 512 tokens. Table 2 shows how a dialogue session is pre-processed.

Temporal difference encoding: In addition to capturing the text contextual information as shown above, we also capture the time difference between multiple turns in case of a multi-turn dialogue. Capturing the time difference is an important factor for IQ prediction as users are likely to immediately interrupt the agent if they do not get the right response. They might even rephrase their request or add more information and hope the agent can take the action they expected in the follow-up turn.

Table 2: Pre-processing of a dialogue session

Timestamp (in seconds)	Dialogue Session
$\tau = 0$	[USER] Play fearless. [AGENT] Playing fearless by Pink Floyd.
$\tau = 2$	[USER] Stop [AGENT] <null>
$\tau = 7$	[USER] Play fearless by Taylor Swift. [AGENT] Here's fearless by Taylor Swift.
Pre-Processed form	
[USER] Play fearless [AGENT] Playing fearless by Pink Floyd [USER] Stop [AGENT] [USER] Play fearless by Taylor Swift [AGENT] Here's fearless by Taylor Swift	

To make the model aware of these temporal differences between the turns, we first select a *reference turn* in the dialogue session and refer to its timestamp as τ_{ref}. This turn is selected at random when pretraining. During fine-tuning, this is the turn for which we have the IQ label. We then calculate the time difference Δ_i for all turns with respect to τ_{ref}: $\Delta_i = \tau_i - \tau_{ref}$, where τ_i is the timestamp of turn i. Δ_i for all turns are then discretized using equal width binning. We create 16 bins to represent equal sized intervals in Δ_i's range of $[-60, 60]$ seconds and map Δ_i to its respective time bin: $[BIN_i]$. The corresponding time-bin embeddings are added to each token of the turn at the input layer of the model, depending on the turn's bin, as shown in Figure 2. These embeddings are learned

from scratch. The number of bins is decided in a way to ensure a uniform distribution of turns across the bins. We reserve a special bin: $[BIN_0]$ for reference turn's tokens. $[BIN_0]$ is the key indicator using which the model recognizes the reference turn.

Task specific heads: As shown in Figure 2, the model has various heads: MLM (Masked Language Modeling) and classifier heads for IQ (Interaction Quality), Domain and Intent classification. During multi-stage pretraining, we only use the MLM head for calculating the loss and updating the weights. The MLM head operates on the output representations of tokens. For multi-task fine-tuning, we use the classification heads for calculating the loss. Each of the classification heads takes the encoded representation of the [CLS] token as input. Each classifier head has a dense layer, followed by a projection layer. All heads are initialized randomly. The output size of the projection layer is equal to the number of labels of the respective task.

Note that we do not design the architecture with real-time/online IQ prediction in mind. We focus on offline evaluation where we have previous and next turns available, when evaluating the quality of the reference turn. However, our design is easily extensible to online evaluation, in which case, the reference turn will always be the last turn in the dialogue session.

3.4 Datasets

Historical Dialogue Sessions: We randomly sample around 2 million English dialogue sessions between users and Alexa from anonymized logged historical data. We do not use any task specific human annotations for these dialogue sessions. These sessions span many NLU domains and intents, and contain turns where the users had both good and bad experiences. As described later, we use these dialogue sessions for the first stage of pretraining.

Interaction Quality (IQ) dataset: This dataset is sampled from Alexa Live Traffic and is annotated with IQ labels provided by experts: 0 (Non-defect or satisfactory experience) and 1 (Defect). Only one turn per dialogue session has a defect/non-defect label, which we refer to as the *reference turn*. The reference turns are labelled from the end user's perspective. For example, considering turn 1 as the reference turn in Table 2, the annotators would give it an IQ label of 1 (defective), as the agent did not play the song intended by the user in that turn. To get the IQ labels, we use a similar **Response Quality annotation workflow** as described in [2]. We have around 500K dialogues for training, 100K for development set, and 100K for testing. The training and test sets are sampled from different time periods. This leads to a test set that has a different domain distribution than the training set, and also some new domains that are not present in the training set. All the turns have domain and intent strings that were produced by a separate NLU system. The IQ prediction task is the primary task at which we want to do better with the least amount of human annotated data possible. For evaluation, we focus on F1-score for the defective class as the binary classification metric. We do so because our dataset is imbalanced (25% defect and 75% non-defect) and identifying dissatisfactory turns is of more importance.

Out-of-domain (OOD) testset: This dataset is sampled from annotated IQ test data. It has 30K instances in total and is only used for evaluation in particular to see the benefits and limitations of pretraining and multi-task learning on out-of-domain instances. To ensure that there is no overlap between the domains/intents of the turns in this testset with the training set, we sample three different fractions (5%, 10% and 25%) from the IQ training dataset (500K instances), and use these subsets for training the models.

3.5 Pretraining: Domain adaptive and Task adaptive

Masked Language Modeling is a common pretraining strategy for transformer based architectures in which a random sample of the tokens in the input sequence is selected and replaced with the special [MASK] token. The MLM loss function is the cross-entropy loss on predicting the masked tokens. Following Liu et al. [13], we conduct token masking dynamically with each batch by masking 15% of the tokens. RoBERTaIQ is initialized from *RoBERTa-base* and is further pretrained as described below. The MLM loss function is defined as:

$$L_{mlm} = -\sum_{m=1}^{M} \log P(x_m) \tag{8}$$

where M is the total number of masked tokens and $P(x_m)$ is the predicted probability of token x_m.

Following [7], we perform the first stage of pretraining on unlabelled historical dialogue sessions data, which we refer to as Domain Adaptive Pretraining (DAPT). Similar to [9], we then further pretrain this model using the MLM loss on the IQ training dataset in the second stage, which we refer to as Task Adaptive Pretraining (TAPT). Note that both DAPT and TAPT do not require any task specific labels.

3.6 Multi-task (MT) fine-tuning

After the multi-stage pretraining process, we finetune the model on the main downstream task of IQ prediction, with additional heads for Domain and Intent prediction. Our hypothesis is that we can benefit from both cross-task data and the regularization effects of MT, especially when the IQ data is small. The multi-task loss is defined in Equation 9:

$$L(\theta) = \sum_{x^i_\psi, y^i_\psi \in D_\psi} \lambda_\psi l(y^i_\psi, f_\psi(Enc_{CLS}(x^i_\psi))) \tag{9}$$

where ψ refers to one of the tasks (IQ, Domain, Intent), x^i, y^i refer to raw dialogue features and task labels respectively, $Enc_{CLS}(x^i_\psi)$ refers to the encoding of $[CLS]$ token after passing x^i_ψ through the shared RoBERTaIQ encoder, f_ψ is the respective task classifier, l is the cross-entropy loss and λ_ψ is the task weight. We empirically set $\lambda_{IQ} = 1$, and $\lambda_{domain} = \lambda_{intent} = 0.5$.

4 EXPERIMENTS

In this section, we first compare RoBERTaIQ with other baselines to see the effects of different model architectures. We then show the results of RoBERTaIQ with multi-stage pretraining and multi-task fine-tuning with varying amounts of IQ training data. All the

experiments were conducted on an AWS p3.16xlarge instance with 8 GPUs. All the numbers reported with "±" prefixes denote absolute differences in the metric w.r.t the corresponding baseline. Other training details and hyperparameters can be found in the Appendix A.

4.1 Comparison with baselines

Table 3: RoBERTaIQ vs baselines on IQ test set (100K examples)

Perf(%)	Accuracy	F1	Precision	Recall
Using 100% IQ training data				
MaUdE++ (Text features only) [20]	86.5	77.4	78.0	76.9
MaUdE++ (+ system metadata)	+ 2.1	+ 1.9	+ 2.7	+ 1.1
RoBERTa-eval [27]	- 3.3	- 5.4	- 5.9	- 5
RoBERTaIQ (This work)	+ 4.2	+ 6.1	+ 6	+6

Table 3 shows the performance comparison between RoBERTaIQ and other baselines on IQ test set. We use the full IQ training data (500K instances) to train all the models. The RoBERTa encoder weights are initialized with a pre-trained model [1] and are finetuned end-to-end with gradients coming from IQ classification loss. We do not apply any multi-tasking or pretraining strategies for this comparison. Using system metadata features helps increase the performance of MaUdE++ by 1.9% F1 score, but RoBERTaIQ, which uses only the text features, still outperforms it by 4.2% absolute F1 score. Please refer to Appendix B for a case study between MaUdE++ and RoBERTaIQ.

RoBERTa-eval performs worse than the MaUdE++ baseline by 5.4% F1 score. This is mainly due to the fact that RoBERTa-eval sees only the left context (previous turns).

4.2 RoBERTaIQ Full Results on IQ testset

Table 4 shows the performance of RoBERTaIQ model on IQ testset with different training settings. To show the respective benefits of pretraining and multi-task learning, we train models with varying amount of IQ training data. "Scratch" refers to training on IQ only, without pretraining or multi-task learning. In DAPT runs, we start with a RoBERTaIQ model pretrained on historical dialogue sessions and finetune on IQ data. In TAPT runs, we start with the DAPT pretrained model, and further pretrain on the IQ training data with MLM loss (without using the IQ labels) and then fine-tune on IQ training data with the classification loss. We perform TAPT experiments only for the cases where we use 100% of available IQ training data. For all the Multi-task runs, we include other classification tasks (Domain and Intent) in addition to IQ for fine-tuning.

Effects of DAPT and TAPT: As can be seen from Table 4, DAPT provides consistent benefits in terms of boosting the performance on IQ prediction task, improving the F1 score by absolute 2.7% in the best case. This successfully demonstrates knowledge transfer from unlabelled historical data to the downstream task of IQ prediction. For the scenario where we use 100% of IQ training data, we see TAPT providing a further boost over DAPT.

[1] https://huggingface.co/roberta-base

Table 4: Model performance comparison with pretraining and finetuning on varying amounts of IQ training data. All the rows show evaluation metrics on IQ testset (100K instances)

Perf (%)	Accuracy	F1	Precision	Recall
IQ (5% Data)				
Scratch (Baseline)	84.7	72.2	75.1	69.5
+ Multi-task	+2.7	+5.2	+4.5	+5.7
+ DAPT	+0.5	+1.8	-1.1	+4.5
+ DAPT + Multi-task	+3.8	**+7.1**	+6.3	+7.5
IQ (10% Training Data)				
Scratch (Baseline)	88.2	78.6	81.6	75.9
+ Multi-task	+0.5	+1.9	-2.3	+5.8
+ DAPT	+1.1	+2.7	-0.2	+5.3
+ DAPT + Multi-task	+1.4	**+2.9**	+0.8	+4.8
IQ (25% Training Data)				
Scratch (Baseline)	89.7	82.1	81.7	82.4
+ Multi-task	0.0	+0.2	-0.6	+1.0
+ DAPT	+0.8	**+1.5**	+0.2	+3.1
+ DAPT + Multi-task	+0.5	+1.0	-0.3	+2.5
IQ (50% Training Data)				
Scratch (Baseline)	90.1	82.2	84.2	80.4
+ Multi-task	+0.2	+0.6	-0.3	+1.5
+ DAPT	+0.6	**+1.9**	-1.8	+5.6
+ DAPT + Multi-task	+0.6	+1.4	-0.1	+2.9
IQ (100% Training Data)				
Scratch (Baseline)	90.7	83.5	84.0	82.9
+ Multi-task	-0.3	-0.3	-0.4	-0.1
+ DAPT	+0.4	+0.7	+0.2	+1.4
+ TAPT	+0.5	**+1.2**	+0.6	+1.8
+ DAPT + Multi-task	+0.1	+0.3	+0.6	+0.1
+ TAPT + Multi-task	+0.1	+0.44	+0.7	+0.3

Effects of Multi-task learning: The performance improvements that come with multi-task learning vary with the IQ training dataset size. We see maximum benefits, i.e. an increase in F1 score by an absolute 5.2% when we use only 5% IQ training dataset. The improvements diminish with increasing IQ training dataset size, even leading to a slight decrease in F1 score when 100% for IQ dataset is used for training. This leads to the conclusion that the additional tasks help the model learn better through extra supervision when IQ training data is small. But after a certain point as the training dataset size increases, cross-task knowledge transfer becomes less useful and instead the regularization effects of other tasks start hurting the performance on the primary task of IQ prediction.

Overall, we find that combining pretraining and multi-task learning provides gains as big as 7.1% F1 score improvement with smaller training dataset sizes. In other words, using these techniques, we can significantly reduce the amount of training dataset (IQ annotations), which is both costly and time-consuming to collect. This can be especially useful for the new or the long-tail domains, for which we do not have IQ annotations available.

4.3 RoBERTaIQ Results on Out-of-Domain (OOD) Dataset

Table 5: Model performance comparison with pretraining and finetuning on varying amounts of IQ training data. All the rows show evaluation metrics on Out-of-domain dataset (30K instances)

Perf (%)	Accuracy	F1	Precision	Recall
IQ (5% Data)				
Scratch (Baseline)	83.71	73.75	73.42	74.09
+ Multi-task	+1.92	+2.12	+5.38	-0.92
+ DAPT	+1.28	+3.33	-0.48	+7.65
+ DAPT + Multi-task	+3.54	**+4.84**	+8.21	+1.69
IQ (10% Data)				
Scratch (Baseline)	88.08	79.28	85.61	73.82
+ Multi-task	-0.30	+0.86	-5.2	+6.05
+ DAPT	+1.97	**+4.32**	-0.42	+8.25
+ DAPT + Multi-task	+0.59	+1.78	-1.75	+4.63
IQ (25% Data)				
Scratch (Baseline)	90.77	84.73	86.64	82.9
+ Multi-task	+.10	+0.49	-1.35	+2.25
+ DAPT	+1.66	**+3.18**	+0.17	+6.13
+ DAPT + Multi-task	+0.99	+1.93	+.02	+3.76

Table 5 shows the evaluation results on the OOD dataset. DAPT consistently provides improvements in F1 scores, irrespective of the size of IQ training dataset used. However, multi-task learning provides improvements only with small IQ training set (5%). The improvements are almost negligible when training dataset size increases, and are much lesser than that of DAPT. This is expected as the knowledge that comes from domain and intent prediction is no longer useful, as we are evaluating on a dataset that has no domains and intents in common with the training dataset. This observation shows the limitations of our multi-task learning setup and emphasizes the importance of choosing domain agnostic auxiliary tasks whose knowledge is transferable to the target task of interest.

5 ABLATION STUDIES

5.1 Contributions of different tasks in Multi-task learning

Here, we gauge the contributions of different tasks that aid in improving the performance on IQ prediction. We conduct the experiments with 5% and 10% IQ training set and other tasks, and remove the tasks one by one to see their impact. As shown in Table 6, the task of Intent prediction is more helpful for improving the performance of IQ prediction. This is expected as there are specific intents that capture user sentiments like pleasantries or insults. Other intents capture instances where the users try to terminate the current turn or exit a skill if the agent is not doing the right thing. These behaviors (as captured by NLU intents) are directly related to user experience and hence get reflected in improved IQ prediction accuracy.

Table 6: Contributions of different tasks in Multi-task learning

Perf (%)	Accuracy	F1	Precision	Recall
IQ (5% Data)				
All tasks (Baseline)	87.4	77.4	79.6	75.2
No Domain	-0.3	-1.1	-1.2	-0.9
No Intent	-1.5	-3	-3.9	-2.1
IQ (10% Data)				
All tasks (Baseline)	88.7	80.5	79.3	81.7
No Domain	-0.2	-0.8	-0.8	-0.8
No Intent	-1.3	-1.7	-2.1	-1.2

5.2 Effect of time encoding in RoBERTaIQ

To see the importance of capturing the time difference between the reference turn and other turns, we do a study where we do not provide any temporal information to the model through the time bin embeddings described in Section 3.3. As shown in Table 7, we see a drop of almost 9% in F1 score if we do not capture the temporal differences between the turns.

Table 7: Effect of time difference encoding in RoBERTaIQ

Perf (%)	Accuracy	F1	Precision	Recall
IQ (100% Data)				
RoBERTaIQ	90.7	83.5	84	82.9
No Time encodings	-5.6	-8.8	-9.1	-8.5

6 CONCLUSION AND FUTURE WORK

In this work, we presented a framework for automatic turn-level dialogue evaluation in large scale Conversational AI systems. We introduce a new modeling architecture on top of RoBERTa that is more suitable for encoding dialogue sessions while capturing the temporal differences across different turns. We showcase that this architecture alone boosts the F1 score by 4.2% over other model architectures used in dialogue evaluation literature, while making use of only the textual features. To make the training more data efficient, we propose multi-stage pretraining and multi-task learning approaches. This helps us leverage large amount of historical dialogue sessions and other system signals, like Domain and Intent, which are readily available. These approaches significantly reduce the requirement of obtaining annotated Interaction Quality (IQ) data. With only 5% of annotated training data, we achieve the same F1 performance as our baseline on IQ testset.

One of the limitations of our multi-task learning approach is that the tasks used are domain specific, which makes it challenging to transfer the knowledge to out-of-domain cases. For future work, we plan to incorporate more auxiliary tasks for multi-task learning that are agnostic to dialogue domains, like user utterance rephrase detection, predicting cohesion between request and response and next session prediction [25]. Pretraining proved to be very promising in improving the IQ prediction performance for both in-domain and out-of-domain cases. In the future, we want to consider much larger scale pretraining by increasing the number of unlabeled historical dialogue sessions by an order of magnitude.

REFERENCES

[1] Praveen Kumar Bodigutla, Aditya Tiwari, Spyros Matsoukas, Josep Valls-Vargas, and Lazaros Polymenakos. 2020. Joint Turn and Dialogue level User Satisfaction Estimation on Multi-Domain Conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 3897–3909. https://www.aclweb.org/anthology/2020.findings-emnlp.347

[2] Praveen Kumar Bodigutla, Longshaokan Wang, Kate Ridgeway, Joshua Levy, Swanand Joshi, Alborz Geramifard, and Spyros Matsoukas. 2019. Domain-Independent turn-level Dialogue Quality Evaluation via User Satisfaction Estimation. *arXiv preprint arXiv:1908.07064* (2019).

[3] Z. Chen, X. Fan, Y. Ling, and C. Guo. 2020. Pre-Training for Query Rewriting in a Spoken Language Understanding System. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 7969–7973. https://doi.org/10.1109/ICASSP40776.2020.9053531

[4] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29, 5 (2001), 1189 – 1232. https://doi.org/10.1214/aos/1013203451

[7] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. *arXiv preprint arXiv:2004.10964* (2020).

[8] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 297–304.

[9] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* (2018).

[10] Weixin Liang, James Zou, and Zhou Yu. 2020. Beyond User Self-Reported Likert Scale Ratings: A Comparison Model for Automatic Dialog Evaluation. *arXiv preprint arXiv:2005.10716* (2020).

[11] Yuan Ling, Benjamin Yao, Guneet Kohli, Tuan-Hung Pham, and Chenlei Guo. 2020. IQ-Net: A DNN Model for Estimating Interaction-level Dialogue Quality with Conversational Agents. In *Proceedings of the KDD 2020 Workshop on Conversational Systems Towards Mainstream Adoption co-located with the 26TH ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD 2020), Virtual Workshop, August 24, 2020 (CEUR Workshop Proceedings, Vol. 2666)*, Giuseppe Di Fabbrizio, Surya Kallumadi, Utkarsh Porwal, and Thrivikrama Taula (Eds.). CEUR-WS.org. http://ceur-ws.org/Vol-2666/KDD_Converse20_paper_12.pdf

[12] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504* (2019).

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[14] Ryan Lowe, Michael Noseworthy, Iulian V. Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2018. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. arXiv:1708.07149 [cs.CL]

[15] Dookun Park, Hao Yuan, Dongmin Kim, Yinglei Zhang, Matsoukas Spyros, Young-Bum Kim, Ruhi Sarikaya, Edward Guo, Yuan Ling, Kevin Quinn, Pham Hung, Benjamin Yao, and Sungjin Lee. 2020. Large-scale Hybrid Approach for Predicting User Satisfaction with Conversational Agents. arXiv:2006.07113 [cs.HC]

[16] Pragaash Ponnusamy, Alireza Roshan Ghias, Chenlei Guo, and Ruhi Sarikaya. 2020. Feedback-Based Self-Learning in Large-Scale Conversational AI Agents. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 08 (Apr. 2020), 13180–13187. https://doi.org/10.1609/aaai.v34i08.7022

[17] Louisa Pragst, Stefan Ultes, and Wolfgang Minker. 2017. Recurrent neural network interaction quality estimation. In *Dialogues with Social Robots*. Springer, 381–393.

[18] Niklas Rach, Wolfgang Minker, and Stefan Ultes. 2017. Interaction quality estimation using long short-term memories. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. 164–169.

[19] Alexander Schmitt and Stefan Ultes. 2015. Interaction quality: assessing the quality of ongoing spoken dialog interaction by experts—and how it relates to user satisfaction. *Speech Communication* 74 (2015), 12–36.

[20] Koustuv Sinha, Prasanna Parthasarathi, Jasmine Wang, Ryan Lowe, William L Hamilton, and Joelle Pineau. 2020. Learning an Unreferenced Metric for Online Dialogue Evaluation. *arXiv preprint arXiv:2005.00583* (2020).

[21] Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2017. RUBER: An Unsupervised Method for Automatic Evaluation of Open-Domain Dialog Systems. arXiv:1701.03079 [cs.CL]

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[23] Marilyn Walker, Candace Kamm, and Diane Litman. 2000. Towards developing general models of usability with PARADISE. *Natural Language Engineering* 6, 3 & 4 (2000), 363–377.

[24] Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 917–929. https://www.aclweb.org/anthology/2020.emnlp-main.66

[25] Ruijian Xu, Chongyang Tao, Daxin Jiang, Xueliang Zhao, Dongyan Zhao, and Rui Yan. 2020. Learning an Effective Context-Response Matching Model with Self-Supervised Tasks for Retrieval-based Dialogues. arXiv:2009.06265 [cs.CL]

[26] Siyang Yuan, Saurabh Gupta, Xing Fan, Derek Liu, Yang Liu, and Chenlei Guo. 2021. Graph Enhanced Query Rewriting for Spoken Language Understanding System. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[27] Tianyu Zhao, Divesh Lala, and Tatsuya Kawahara. 2020. Designing Precise and Robust Dialogue Response Evaluators. arXiv:2004.04908 [cs.CL]

Table 8: Comparison of RoBERTaIQ model predictions with MaUdE++. Reference turn in each session is boldfaced.

#	Timestamp (in seconds)	Dialogue Session	IQ label	MaUdE++ prediction	RoBERTaIQ prediction
1	τ = 0	**[USER] how much do amungus plushies cost** **[AGENT] One moment, for amungus plushies this is the closest I could find. The first one is Winsterch Kids Cats Stuffed Animal Toys**	1	0	1
	τ = 9	[USER] how much do among us plushies cost [AGENT] One moment. Showing results for among us plushies. The first one is Bedtime Originals Among Us Plush Toy.			
2	τ = 0	[USER] Start joke responder [AGENT] Ok, Here's Joke Responder Welcome to joke responder. Please tell me a joke.	0	1	0
	τ = 6	**[USER] why did the skeleton not cross the road** **[AGENT] I don't know, why did the skeleton not cross the road**			
	τ = 11	[USER] because he didn't have the guts! [AGENT] Ha ha ha. So much talent! So many laughs!			
3	τ = 0	**[USER] show me shark videos** **[AGENT] Here's what I found (*playing video*)**	1	0	1
	τ = 8	[USER] play baby shark on amazon prime [AGENT] Here's Baby Shark , by Pinkfong , on Amazon Music.			

A TRAINING SETUP

We use the HuggingFace Transformers [2] library for all our training and evaluation code. Table 9 and 10 show the hyperparameters for pretraining and fine-tuning, respectively.

Table 9: Hyperparameters for Pretraining (DAPT and TAPT)

Hyperparameter	Assignment
maximum epochs	20
MLM masking probability	0.15
max learning rate	5e-4
optimizer	AdamW
batch size per GPU	4
gradient accumulation steps	32
learning rate decay	Linear
effective batch size	4*32*8 = 1024
Adam epsilon	1e-6
Adam betas	0.9, 0.999

Table 10: Hyperparameters for Fine-tuning

Hyperparameter	Assignment
maximum epochs	20
patience (for early stopping)	4
max learning rate	1e-5
dropout	0.1
optimizer	AdamW
batch size per GPU	4
gradient accumulation steps	8
learning rate decay	Linear
effective batch size	4*8*8 = 256
Adam epsilon	1e-6
Adam betas	0.9, 0.999

[2]https://github.com/huggingface/transformers

B CASE STUDY

Table 7 shows a comparison of model predictions of RoBERTaIQ with MaUdE++ . We use the better performing version of MaUdE++, i.e., the one that also uses system metadata as input features.

Dialogue 1: By looking at turn 1 (reference turn) in isolation, it appears that the agent gave the right response. However, turn 2 makes it clear that turn 1 had an ASR error; "Among Us" was incorrectly recognized as "amungus". RoBERTaIQ's encoder was able to recognize this, as it has the ability to apply self-attention mechanism across tokens of different turns, and hence correctly predicted IQ score = 1, for turn 1.

Dialogue 2: In turn 2, the user is interacting with a "Joke Insider" 3P skill. MaUdE++ predicts that the response is unsatisfactory due to presence of "*I don't know*". However, RoBERTaIQ made a better sense of the response using the previous and next turns, and predicted the response to be satisfactory (IQ score = 0).

Dialogue 3: MaUdE++ failed to recognize from the context that the user wanted to listen to *Baby Shark*, while RoBERTaIQ figured that out and correctly predicted IQ score = 1 as the agent showed the user irrelevant shark videos in turn 1.