Using Interactive Feedback to Improve the Accuracy and Explainability of Question Answering Systems Post-Deployment

Anonymous ACL submission

Abstract

Most research on question answering focuses on the pre-deployment stage; i.e., building an accurate model for deployment. In this paper, we ask the question: Can we improve QA systems further *post*-deployment based on user interactions? We focus on two kinds of improvements: 1) improving the QA system's performance itself, and 2) providing the model with the ability to explain the correctness or incorrectness of an answer. We collect a retrieval-based QA dataset, FEEDBACKQA, which contains interactive feedback from users. We collect this dataset by deploying a base QA system to crowdworkers who then engage with the system and provide feedback on the quality of its answers. The feedback contains both structured ratings and unstructured natural language explanations. We train a neural model with this feedback data that can generate explanations and re-score answer candidates. We show that usage of the feedback data improves the accuracy of the QA system, and helps users make informed decisions about the correctness of answers.[1]

1 Introduction

Much of the recent excitement in question answering (QA) is in building high-performing models with carefully curated training datasets. Datasets like SQuAD (Rajpurkar et al., 2016), NaturalQuestions (Kwiatkowski et al., 2019) and CoQA (Reddy et al., 2019) have enabled rapid progress in this area. Most existing work focuses on the pre-deployment stage; i.e., training the best QA model before it is released to users. However, this stage is only one stage in the potential lifecycle of a QA system.

In particular, an untapped resource is the large amounts of user interaction data produced after the initial deployment of the system. Gathering this data should in practice be relatively cheap, since top QA systems are of high enough quality that users would genuinely use them to seek information.

Exploiting this user interaction data presents new research challenges, since they typically consist of a variety of weak signals. For example, user clicks could indicate answer usefulness (Joachims, 2002), users could give structured feedback in the form of ratings to indicate the usefulness (Stiennon et al., 2020), or they could give unstructured feedback in natural language explanations on why an answer is correct or incorrect. User clicks have been widely studied in the field of information retrieval (Joachims, 2002). Here we study the usefulness of *interactive feedback* in the form of ratings and natural language explanations.

Whilst there are different variants of QA tasks, this paper focuses primarily on retrieval-based QA (RQA; Chen et al. 2017; Lee et al. 2019). Given a question and a set of candidate answer passages, a model is trained to rank the correct answer passage the highest. In practice, when such a system is deployed, an user may engage with the system and provide feedback about the quality of the answers. Such feedback is called interactive feedback. Due to the lack of a dataset containing interactive feedback for RQA, we create FEEDBACKQA.

FEEDBACKQA is a large-scale English QA dataset containing interactive feedback in two forms: user ratings (structured) and natural language explanations (unstructured) about the correctness of an answer. Figure 1 shows an example from FEEDBACKQA. The dataset construction has two stages: We first train a RQA model on the questions and passages, then deploy it on a crowdsourcing platform. Next, crowdworkers engage with this system and provide interactive feedback. To make our dataset practically useful, we focus on question answering on public health agencies for the Covid-19 pandemic. The base model for FEEDBACKQA is built on 28k questions and 3k passages from various agencies. We collect 9k interactive

[1] We will make both the data and the code public.

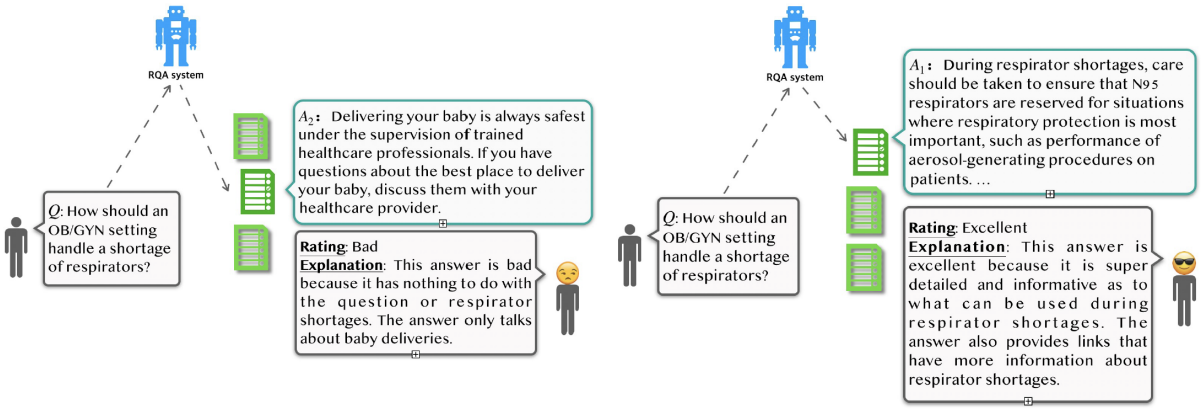

Figure 1: Users interact with the deployed QA model and give feedback. Feedback contains a rating (*bad, good, could be improved, excellent*) and a natural language explanation.

feedback data samples for the base model.

We investigate the usefulness of the feedback for improving the RQA system in terms of two aspects: answer accuracy and explainability. Specifically, we are motivated by two questions: 1) Can we improve the answer accuracy of RQA models by learning from the interactive feedback? and 2) Can we learn to generate explanations that help humans to discern correct and incorrect answers?

To address these questions, we use feedback data to train models that rerank the original answers as well as provide an explanation for the answers. Our experiments show that this approach not only improves the accuracy of the base QA model for which feedback is collected but also other strong models for which feedback data is not collected. Moreover, we conduct human evaluations to verify the usefulness of explanations and find that the generated natural language explanations help users make informed and accurate decisions on accepting or rejecting answer candidates.

Our contributions are two-fold:

1. We create the first retrieval-based QA dataset containing interactive feedback.

2. We propose a simple and effective method of using the feedback data to increase the accuracy and explainability of RQA systems.

2 FEEDBACKQA Dataset

Recently, there have been efforts to collect feedback data in the form of explanations for natural language understanding tasks (Camburu et al. 2018; Rajani et al. 2019, *inter alia*). These contain explanations only for ground-truth predictions for a given input sampled from the training data without any user-system interaction. Instead, we collect user feedback after deploying a RQA system thereby collecting feedback for both correct and incorrect predictions. Table 1 presents a comprehensive comparison of FEEDBACKQA and existing natural language understanding (NLU) datasets with explanation data.

2.1 Dataset collection

In order to collect post-deployment feedback as in a real-world setting, we divide the data collection into two stages: pre-deployment (of a RQA model) and post-deployment.

Stage 1: Pre-deployment of a QA system We scrape Covid-19-related content from the official websites of WHO, US Government, UK Government, Canadian government,[2] and Australian government. We extract the questions and answer passages in the FAQ section. In addition, we clean the scraped pages and extract additional passages for which we curate corresponding questions using crowdsourcing. We present additional details on this annotation process in Appendix A. We use this dataset to train a base RQA model for each source separately and deploy them. For the base model, we use a BERT-based dense retriever (Karpukhin et al., 2020) combined with Poly-encoder (Miller et al., 2017) (more details are in Section 3.1).

Stage 2: Post-deployment of a QA system Since each domain has several hundred passages (Table 2), it is hard for a crowdworker to ask questions that cover a range of topics in each source. We thus collect questions for individual passages beforehand similar to Stage 1 and use these as in-

[2]We focus on the Province of Quebec

Datasets	Task	Feedback Type	Interactive Feedback	Feedback for incorrect predictions
e-SNLI (Camburu et al., 2018)	NLI	Free-form	✗	✗
CoS-E (Rajani et al., 2019)	Commonsense QA	Free-form	✗	✗
LIAR-PLUS (Alhindi et al., 2018)	Fact checking	Free-form	✗	✗
QED (Lamm et al., 2021)	Reading comprehension	Structured	✗	✗
NExT (Wang et al., 2019)	Text classification	Structured	✗	✗
FEEDBACKQA	Retrieval-based QA	Structured & Free-form	✓	✓

Table 1: Comparison of FEEDBACKQA with existing NLU datasets containing feedback in the form of structured representations (according to a schema) or natural language explanations (free-form).

	#Passages	#Questions	#Feedback
Australia	584	1783	2264
Canada	587	8844	/
UK	956	2874	3668
US	598	13533	2628
WHO	226	688	874
Overall	2951	27722	9434

Table 2: Number of samples in different domains of FEEDBACKQA. We split the data into train/validation/test sets in the ratio of $0.7 : 0.1 : 0.2$.

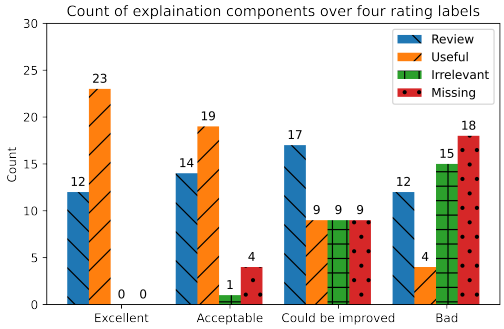

Figure 2: Distribution of component number in 100 natural language feedback of different rating labels.

teractive questions. The question and top-2 predictions of the model are shown to the user and they give feedback for each question-answer pair. The collected feedback consists of a rating, selected from *excellent, good, could be improved, bad*, and a natural language explanation elaborating on the strengths and/or weaknesses of the answer. For each QA pair, we elicit feedback from three different workers. We adopted additional strategies to ensure the quality of the feedback data, the details of which are available in Appendix B. The resulting dataset statistics are shown in Table 2. In order to test whether interactive feedback also helps in out-of-distribution settings, we did not collect feedback for one of the domains (Canada).

2.2 FEEDBACKQA analysis

Table 3 shows examples of the feedback data, including both ratings and explanations. We find that explanations typically contain review-style text indicating the quality of the answer, or statements summarizing which parts are correct and why. Therefore, we analyze a sample of explanations using the following schema:

Review Several explanations start with a generic review such as *This directly answers the question* or *It is irrelevant to the question.* Sometimes users also highlight aspects of the answer that are good or can be improved. For instance, ... *could improve grammatically* ... suggests that the answer could be improved in terms of writing.

Summary of useful content refers to the part of answer that actually answers the question;

Summary of irrelevant content points to the information that is not useful for the answer, such as off-topic or addressing incorrect aspects;

Summary of missing content points the information the answer fails to cover.

We randomly sample 100 explanations and annotate them. Figure 2 shows the distribution of the types present in explanations for each rating label. All explanations usually contain some review type information. Whereas explanations for answers labeled as excellent or acceptable predominantly indicate the parts of the answer that are useful. The explanations for answers that can be improved indicate parts that are useful, wrong or missing. Whereas bad answers often receive explanations that highlight parts that are incorrect or missing as expected.

3 Experimental Setup

FEEDBACKQA contains two types of data. One is pre-deployment data $\mathcal{D}_{\text{pre}} = (Q, A^+, \mathcal{A})$, where

Rating label	Explanation
Excellent	This answers the question directly. This answer provides information and recommendation on how people and adolescent can protect themselves when going online during the Covid-19 pandemic.
Acceptable	This answer, while adequate, could give more information as this is a sparse answer for a bigger question of what one can do for elderly people during the pandemic.
Could be improved	The answer relates and answers the question, but could improve grammatically and omit the "yes"
Could be improved	The answer is about some of the online risks but not about how to protect against them.
Bad	This does not answer the question. This information is about applying visa to work in critical sector. It does not provide any information on applying for Covid-19 pandemic visa event as asked in the question.

Table 3: Examples of explanation and its associated rating label. Span color and their types of components: generic and aspect review ; summary of useful content ; summary of irrelevant content ; summary of missing content

Q is a question paired with its gold-standard answer passage A^+ from the domain corpus $\mathcal{A}$. The other is post-deployment feedback data $\mathcal{D}_{\text{feed}} = (Q, A, Y, E)$, where Q is a question paired with a candidate answer $A \in \mathcal{A}$ and corresponding feedback for the answer. The feedback consists of a rating Y and an explanation E. We build two kinds of models on pre- and post-deployment data: RQA models on the pre-deployment data that can retrieve candidate answers for a given question, and feedback-enhanced RQA models on the post-deployment data that can rate an answer for a given question as well as generate an explanation for the answer. We use this rating to rerank the answer candidates. Therefore, in our setting, a feedback-enhanced RQA model is essentially a *reranker*. Keeping in mind the fact that real-world QA systems evolve quickly, we decouple the reranker model from the RQA model by using separate parameters for the reranker independent of the RQA model. We train this reranker on the feedback data. This allows for the reranker to be reused across many RQA models. We leave other ways to enhance RQA models with feedback data for future work. Below, we describe the architectures for the RQA models and feedback-based rerankers.

3.1 RQA Models (Pre-deployment)

We use dense passage retrievers (Karpukhin et al., 2020) to build the RQA models, where the similarity between the question embedding and the passage embedding is used to rank candidates. We use two variants of pre-trained models to obtain the embeddings: 1) BERT (Devlin et al., 2019), a pre-trained Transformer encoder; and 2) BART (Lewis et al., 2020), a pretrained Transformer encoder-decoder. For BERT, we use average pooling of token representations as the embedding, whereas for BART we use the decoder's final state. While Karpukhin et al. use question-agnostic passage representations, we use a poly-encoder (Humeau et al., 2020) to build question-sensitive document representations. In a poly-encoder, each passage is represented as multiple encodings, first independent of the question, but then a simple attention between the question and passage embeddings is used to compute question-sensitive passage representation, which is later used to compute the relevance of the passage for a given query. Humeau et al. show that the poly-encoder architecture is superior to alternatives like the bi-encoder (Karpukhin et al., 2020) without much sacrifice in computational efficiency.[3]

Given pre-deployment training data $\mathcal{D}_{\text{pre}} = (Q, A^+, \mathcal{A})$, the RQA model parameterized by θ is trained to maximize the log-likelihood of the correct answer:

$$\mathcal{J}_\theta = \log P_\theta(A^+|Q, \mathcal{A})$$
$$P_\theta(A^i|Q, \mathcal{A}) = \frac{\exp(S(Q, A^i))}{\sum_{A \in \mathcal{A}} \exp(S(Q, A))} \quad (1)$$

Here $S(Q, A)$ denotes the dot product similarity between the question and passage embedding. As it is inefficient to compute the denominator over all passages during training, we adopt an in-batch negative sampling technique (Humeau et al., 2020), merging all of the A^+ in the same minibatch into a set of candidates.

3.2 Feedback-enhanced RQA models (Post-deployment)

On the post-deployment data $\mathcal{D}_{\text{feed}} = (Q, A, Y, E)$, we train a reranker that assigns a rating to an answer

[3]The performance results of poly-encoder and bi-encoder for our task are shown in Table 9.

and also generates an explanation. We use BART parameterized by ϕ as the base of EXPLAINRATE because it is ease to adapt it to both explanation generation and rating classification. The encoder of the BART model takes as input the concatenation $[Q; \text{SEP}; A]$, and the decoder generates an explanation E; after that, an incremental fully-connected network predicts the rating Y given the last hidden states of decoder. The rating is used to score QA pairs, whereas the generated explanation is passed to humans to make an informed decision of accepting the answer. We also implement a variant where the model directly produces a rating without generating an explanation. Since each candidate answer is annotated by different annotators, an answer could have multiple rating labels. To account for this, we minimize the KL-divergence between the the target label distribution and the predicted distribution:

$$\mathcal{J}_{\phi'} = -D_{\text{KL}}(P(Y|Q,A)||P_\phi(Y|Q,A)),$$
$$P(Y_i = y|Q_i, A_i) = \frac{C_{y,i}}{\sum_y C_{y,i}} \quad (2)$$

where $C_{y,i}$ is the count of the rating label y for the i-th feedback.

In order to enhance an RQA model with the reranker, we first select the top-k candidates according to the RQA model (in practice we set $k = 5$). The reranker then takes as input the concatenation of the question and each candidate, then generates a rating for each answer. We simply sum up the scores from the RQA model and the reranker model. In practice, we found that using the reranker probability of *excellent* worked better than normalizing the expectation of the rating score (from score 0 for label *bad* to 3 for *excellent*). So, we score the candidate answers as follows:

$$S(A|\mathcal{A}, Q) = P_\theta(A = A^+|\mathcal{A}, Q) + P_\phi(y = \textit{excellent}|A, Q) \quad (3)$$

4 Experiments and Results

We organize the experiments based on the following research questions:

- RQ1: Does feedback data improve the base RQA model accuracy?
- RQ2: Does feedback data improve the accuracy of RQA models that are stronger than the base model?
- RQ3: Do explanations aid humans in discerning between correct and incorrect answers?

We answer these questions by comparing the RQA models with the feedback-enhanced RQA models. The implementation and hyper-parameter details of each model are included in Appendix D.

4.1 RQ1: Does feedback data improve the base RQA model?

Model details. Our base model is a BERT RQA model which we deployed to collect feedback data to train the other models (Section 3.1).

For the feedback-enhanced RQA model, we use the BART-based reranker described in Section 3.2. We train one single model for all domains. We call this FEEDBACKRERANKER. We compare two variants of FEEDBACKRERANKER on validation set, one of which directly predicts the rating while the other first generates an explanation and then the rating. And we found the first one performs slightly better (Appendix Table 10). We conjecture that learning an explanation-based rating model from the limited feedback data is a harder problem than directly learning a rating model. Therefore, for this experiment, we only use the rating prediction model (but note that explanation-based rating model is already superior to the base RQA model).

To eliminate the confounding factor of having a larger number of model parameters introduced by the reranker, we train another reranker model on the pre-deployment data VANILLARERANKER and compare against the reranker trained on the feedback data. To convert the pre-deployment data into the reranker's expected format, we consider a correct answer's rating label to be *excellent*, and the randomly sampled answer candidates[4] to be *bad*. Note that this dataset is much larger than the feedback data.

Finally, we combine the training data of FEEDBACKRERANKER and VANILLARERANKER and train the third reranker called COMBINEDRERANKER.

To measure retrieval accuracy, we adopt Precision@1 (P@1) as our main metric.

Results. As shown in Table 4, the feedback-enhanced RQA model is significantly[5] better than the base RQA model by 1.84 points. Although VANILLARERANKER improves upon the base model, it is weaker than FEEDBACKRERANKER,

[4]We also tried using the top predictions from the base QA model, but found this approch leads to slightly worse performance than negative sampling.

[5]We follow Berg-Kirkpatrick et al. (2012) to conduct the statistical significant test

Methods	Australia	US	Canada	UK	WHO	All	Beats
BERT RQA model ✦	47.25	65.30	81.49	48.50	81.19	64.75	None
+ FEEDBACKRERANKER ✻	55.13	65.97	83.74	51.07	77.05	66.59	✦❁
+ VANILLARERANKER ❁	54.29	64.80	83.20	49.63	77.96	65.98	✦
+ COMBINEDRERANKER ♦	**55.63**	**67.54**	**84.99**	**53.21**	**78.51**	**67.97**	✦✻❁

Table 4: Accuracy of the BERT RQA model, i,.e., the deployed model, and its enhanced variants on the test set. FEEDBACKRERANKER is trained on the post-deployment feedback data, VANILLARERANKER is trained on the pre-deployment data and COMBINEDRERANKER is trained on both. The column *Beats* indicates that the model significantly outperforms (p-value < 0.05) the competing methods. All of the results are averaged across 3 runs.

Methods	Australia	US	Canada	UK	WHO	All	Beats
BART RQA model ♈	52.88	68.47	82.49	51.29	81.97	67.42	None
+ FEEDBACKRERANKER ♉	54.78	70.45	84.38	53.47	82.51	69.12	♈♊
+ VANILLARERANKER ♊	53.09	70.40	82.76	53.08	82.33	68.33	♈
+ COMBINEDRERANKER ♣	**55.27**	**71.45**	**85.35**	**54.83**	**83.61**	**70.10**	♈♉♊

Table 5: Accuracy of the BART RQA model and its enhanced variants on the test set. All of the results are averaged across 3 runs.

and COMBINEDRERANKER is a much stronger model than any of the models, indicating that learning signals presented in feedback data and the pre-deployment data are complementary to each other. Moreover, we also see improved performance on the Canada domain, although feedback data was not collected for that domain.

From these experiments, we conclude that feedback data can improve the accuracy of the base RQA model, not only for the domains for which feedback data is available but also for unseen domains (Canada).

4.2 RQ2: Does feedback data improve the accuracy of RQA models that are stronger than the base model?

If feedback data were only useful for the base RQA model, then its usefulness would be questionable, since the RQA development cycle is continuous and the base RQA model will eventually be replaced with a better model. For example, we find that BART-based dense retriever is superior than the BERT RQA model: Table 9 in Appendix E shows the results on validation set which indicate that BART RQA model overall performance is nearly 4 points better than the BERT RQA model.

To answer RQ2, we use the same FEEDBACKRERANKER and VANILLARERANKER to rescore the BART RQA predictions, even though feedback data is not collected for this model. We observe that the resulting model outperforms the BART RQA model in Table 5, indicating that the feedback data is still useful. Again, FEEDBACKRERANKER is superior to VANILLARERANKER although the feedback data has fewer samples than the pre-deployment data, and the COMBINEDRERANKER has the best performance.

These results suggest that the feedback data is useful not only for the base RQA model but also other stronger RQA models.

4.3 RQ3: Do explanations aid humans in discerning between correct and incorrect answers?

We conduct a human evaluation to investigate whether explanations are useful from the perspective of users. Unfortunately, rigorous definitions and automatic metrics of explainability remain open research problems. In this work, we simulate a real-world scenario, where the user is presented an answer returned by the system as well as an explanation for the answer, and they are asked to determine whether the answer is acceptable or not. Jacovi and Goldberg (2020) advocate utility metrics as proxies to measure the usefulness of explanations instead of directly evaluating an explanation since plausible explanations does not necessarily increase the utility of the resulting system. Inspired by their findings, we measure if explanations can: 1) help users to make accurate decisions when judging an answer (with respect to a ground truth) and 2) improve the agreement among users in accepting/rejecting an answer candidate. The former measures the utility of an explanation and

Explanation	Accuracy	Agreement
Blank	69.17	0.31
Human-written	88.33	0.80
BART feedback model	81.67	0.71
BART summarization model	74.17	0.30

Table 6: Human evaluation results of the usefulness of explanations. Accuracy measures the utility of explanations in selecting the correct rating label for an answer, whereas agreement measures whether explanations invoke same behaviour pattern across users.

the latter measures if the explanations invoke the same behavioral pattern across different users irrespective of the utility of the explanation. Note that agreement and utility are not tightly coupled. For example, agreement can be higher even if the utility of an explanation is lower when the explanation misleads end users to consistently select a wrong answer (González et al., 2021; Bansal et al., 2021).

We sample 60 feedback samples from the hidden split of the feedback data $\mathcal{D}_{\text{feed}} = (Q, A, Y, E)$ for evaluation purposes.[6] We evaluate four experimental setups on these samples which vary in the type of explanation shown to the end users: 1) no explanation; 2) human-written explanations; 3) explanations generated by the BART model trained on the feedback data (Section 3.2); and 4) summary of the answer candidate generated by a strong fine-tuned BART-based summarization model.[7] The last setting is inspired from the observation in Section 2.2 that a large portion of explanations contain summary of questions/answers. We investigate if conventional summary of an answer is as useful as an explanation. For each of these setups, two crowdworkers assign a rating label to each answer candidate indicating the quality of the answer. Each setup has its own set of workers in order to avoid information-leakage across setups (this simulates A/B testing often used by production systems).

We measure the workers' accuracy (average of the two workers) in determining the correctness of an answer with respect to the original annotation in FEEDBACKQA, as well as compute the agreement of workers with each other using Spearman corre-

[6]For simplicity, we merge the answer feedback labels *good* and *could be improved* into one label called *partially correct*. The final set of answer labels we used are *correct* (for original label *excellent*), *partially correct* and *incorrect* (for original label *bad*).

[7]https://huggingface.co/facebook/bart-large-xsum

Question: *What are the restrictions to travelers landing in the US?*
Answer: *... Under current federal regulations, pilots must report all illnesses and deaths to CDC before arriving to a US destination. According to CDC disease protocols, if a sick traveler is considered to be a public health risk, CDC works with local and state health departments and international public health agencies to contact passengers and crew exposed to that sick traveler. Be sure to give the airline your current contact information when booking your ticket so you can be notified if you are exposed to a sick traveler on a flight. ...*
Generated explanation: *This answer is irrelevant to the question. The question is asking about restrictions on travelers landing in the US and this answer provides information on what to do if there is a sick passenger on an international or domestic flight.*
Human explanation: *This information does not answer the question. The question is asking for information on the restrictions to travelers landing in the US and this answer contains information on pilots reports on illnesses and deaths before arriving to a US destinations and airline notification when passengers are exposed to sick travelers on a flight.*
Generated Summary: *The Centers for Disease Control and Prevention (CDC) is working with airlines to help prevent illness and death on flights.*
Question: *Should I wear gloves when I go out in public to prevent Covid-19?*
Answer: *No, WHO does not recommend the regular use of gloves by people in the community. The wearing of gloves may increase risks, since it can lead to self-infection or transmission to others through touching contaminated surfaces which is not followed by removal of gloves and hand hygiene. Therefore, in public places such as supermarkets, in addition to physical distancing, WHO recommends the installation of public hand hygiene stations at the entrance and exit...*
Generated Explanation: *This answer is excellent because it unequivocally states that gloves are not recommended. Therefore, it answers the question directly. It goes on to explain why gloves aren't recommended. This is a great answer.*
Human explanation: *The question is answered directly. The answer states not to wear gloves in public.*
Generated Summary: *The World Health Organization (WHO) has issued a guidance on the use of gloves in public places.*
Question: *Can students work in aged care?*
Answer:*...The Australian Government has announced that residential aged care facilities should take extra precautions to keep residents safe from COVID-19. Both individuals and management need to take responsibility for the health of visitors and staff at facilities to protect our most vulnerable community members. Aged care facilities will: closely monitor the health of staff...*
Generated Explanation: *This answer does not address whether or not students are allowed to work in aged care. This answer only provides information on precautions to be taken by aged care facilities to keep residents safe.*
Human Explanation: *This answer is about covid and animals. The question was about getting the virus from a pet.*
Generated Summary: *Residents in aged care facilities across Australia are being urged to take extra precautions to prevent the spread of a deadly virus.*

Table 7: Examples of different explanation types: model-generated and human-written explanation and model-generated summary.

lation. Table 6 presents the results. All explanation types improve accuracy compared to the the model with no explanations. This could be because any explanation forces the worker to think more about an answer. The human-written explanations has the highest utility and also leads to the biggest agreement. Both the human-written explanations and the explanations generated by the BART feedback model have more utility and higher agreement than the BART summarization model. In fact, the summarization model leads to lower agreement than showing no explanation.

These results indicate that explanations based on feedback data are useful for end users in discerning correct and incorrect answers, and they also improve the agreement across users.

Table 7 shows some examples of explanation that helps the users make more informed and accurate decision. In the first example, the model-generated explanation points out the gap between the question and the answer candidate, though there are a large number of overlapping keywords. Meanwhile, human-written explanations are generally more abstractive and shorter in nature (e.g., see the second example).

5 Related work

Retrieval-based question answering has been widely studied, from early work on rule-based systems (Kwok et al., 2001), to recently proposed neural-based models (Yang et al., 2019; Karpukhin et al., 2020). Most existing work focuses on improving the accuracy and efficacy by modification of a neural architecture (Karpukhin et al., 2020; Humeau et al., 2020), incorporation of external knowledge (Ferrucci et al., 2010), and retrieval strategy (Kratzwald and Feuerriegel, 2018). These methods focus on the pre-deployment stage of RQA models.

By contrast, we investigate methods to improve a RQA model post-deployment with interactive feedback. The proposed methods are agnostic to the architecture design and training methods of the base RQA model.

Learning from user feedback has been a long standing problem in natural language processing. Whilst earlier work proposes methods for using implicit feedback—for instance, using click-through data for document ranking (Joachims, 2002)—recent work has explored explicit feedback such as explanations of incorrect responses by chatbots (Li et al., 2016; Weston, 2016) and correctness labels in conversational question answering and text classification (Campos et al., 2020). However, the feedback in these studies is automatically generated using heuristics, whereas our feedback data is collected from human users. Hancock et al. (2019) collect suggested responses from users to improve a chatbot, while we investigate the effect of natural feedback for RQA models.

Explainability and Interpretability has received increasing attention in the NLP community recently. This paper can be aligned to recent efforts in collecting and harnessing explanation data for language understanding and reasoning tasks, such as natural language inference (Camburu et al., 2018; Kumar and Talukdar, 2020), commonsense question answering (Rajani et al., 2019), document classification (Srivastava et al., 2017), relation classification (Murty et al., 2020), reading comprehension (Lamm et al., 2021), and fact checking (Alhindi et al., 2018). The type of feedback in FEEDBACKQA differs from the existing work in several aspects: 1) FEEDBACKQA has feedback data for both positive and negative examples, while most of other datasets only contains explanations of positive ones; 2) FEEDBACKQA has both structured and unstructured feedback, while previous work mainly focuses on one of them; 3) The feedback in FEEDBACKQA is collected post-deployment; 4) While previous work aims to help users interpret the model decisions with natural language explanation, we investigate whether feedback-based explanations increase the utility of the deployed system.

6 Conclusion

In this work, we investigate the usefulness of feedback data in retrieval-based question answering. We collect a new dataset FEEDBACKQA, which contains interactive feedback in the form of ratings and natural language explanations. We propose a method to improve the RQA model with the feedback data, training a reranker to select an answer candidate as well as generate the explanation. We find that this approach not only increases the accuracy of the deployed model but also other stronger models for which feedback data is not collected. Moreover, our human evaluation results show that both human-written and model-generated explanations help users to make informed and accurate decisions about whether to accept an answer.

References

Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 85–90. Association for Computational Linguistics.

Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. 2021. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural Language Inference with Natural Language Explanations. In *Advances in Neural Information Processing Systems 31*, pages 9539–9549.

Jon Ander Campos, Kyunghyun Cho, Arantxa Otegi, Aitor Soroa, Eneko Agirre, and Gorka Azkune. 2020. Improving conversational question answering systems after deployment using feedback-weighted learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2561–2571.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.

Ana Valeria González, Gagan Bansal, Angela Fan, Yashar Mehdad, Robin Jia, and Srinivasan Iyer. 2021. Do explanations help users detect errors in open-domain QA? an evaluation of spoken vs. visual explanations. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1103–1116.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Transformer Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. *arXiv:1905.01969 [cs]*. ArXiv: 1905.01969.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205. Association for Computational Linguistics.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *SIGKDD*. Association for Computing Machinery.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Bernhard Kratzwald and Stefan Feuerriegel. 2018. Adaptive document retrieval for deep question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 576–581.

Sawan Kumar and Partha Talukdar. 2020. Nile: Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Cody CT Kwok, Oren Etzioni, and Daniel S Weld. 2001. Scaling question answering to the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 150–161.

Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. 2021. Qed: A framework and dataset for explanations in question answering. *Transactions of the Association for Computational Linguistics*, 9:790–806.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2016. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*.

Alexander H Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. Parlai: A dialog research software platform. In *EMNLP (System Demonstrations)*.

Shikhar Murty, Pang Wei Koh, and Percy Liang. 2020. Expbert: Representation engineering with natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2106–2113.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Shashank Srivastava, Igor Labutov, and Tom Mitchell. 2017. Joint concept learning and semantic parsing from natural language explanations. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1527–1536.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021.

Ziqi Wang, Yujia Qin, Wenxuan Zhou, Jun Yan, Qinyuan Ye, Leonardo Neves, Zhiyuan Liu, and Xiang Ren. 2019. Learning from explanations with neural execution tree. In *International Conference on Learning Representations*.

Jason E Weston. 2016. Dialog-based language learning. *Advances in Neural Information Processing Systems*, 29:829–837.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77.

A Details of Data Collection

Passage curating After we scraped the websites, we collect the questions and answers in the Frequently-Asked-Questions pages directly. For those pages without explicit questions and answers, we extract the text content as passages and proceed to question collection.

Question collection We hire crowd-source workers at the Amazon MTurk platform to write questions conditioned on the extracted passages. The workers are instructed not to ask too generic questions or copy and paste directly from the passages.

A qualification test with two sections is done to pick up the best performing workers. In the first section, the workers are asked to distinguish the good question from the bad ones for given passages. The correct and incorrect questions were carefully designed to test various aspects of low-quality submissions we had received in the demo run. The second section is that writing a question given a passage. We manually review and score the questions. We paid 0.2$ to workers for each question.

B Details of Feedback Collection

We asked the workers to provide rating and natural language feedback for question-answer pairs. For qualification test, we labeled the rating for multiple pairs of questions and answers. The workers are selected based on their accuracy of rating labeling. We paid 0.4$ to workers for each feedback.

C Details of Human Evaluation

The worker assignment is done to make sure a worker rates the same question-answer pair only once. Otherwise there is risk that the workers just blindly give the same judgement for a certain QA pair.

We adopt the qualification test similar to the one for feedback collection. We also include some dummy QA pairs, whose answer candidate were randomly sampled from the corpora, and we filter out the workers who fail to recognize them. We paid 0.3$ to workers for each QA pair.

D Implementation Details

Throughout the experiments, we have used 4 32-GB Nvidia Tesla V100. The hyperparameter (learning rate, dropout rate) optimisation is performed for the RQA models only and standard fine-tuning hyperparameters of BART are used for building the FEEDBACKRERANKER model. We set batch size as 16. We truncate the questions and passages to 50 and 512 tokens, respectively. The models are trained with 40 epochs. For our hyperparameter search, we have used 5 trials and while reporting the final results the best hyperparameter variant's performance was averaged across 3 different runs. All experiment runs were finished within 20 hours.

	lr	Dropout
BERT (Bi-encoder)	5.0e-05	0.1
BERT (Poly-encoder)	5.0e-05	0.1
BART (Bi-encoder)	9.53e-05	0.01026
BART (Poly-encoder)	4.34e-05	0.1859
FEEDBACKRERANKER	5.0e-05	0.1

Table 8: Hyper-parameter setting of different variants of QA models as well as EXPLAINRATE and RATEONLY. There is no pooling operation in the latter two models.

E Validation performance

In addition to the Poly-encoders, we also explore Bi-encoder and we have found that its performance is consistently worse. Table 9 presents the performance of base QA models with different pre-trained Transformer models and encoding methods on the validation set.

Methods	Australia	US	Canada	UK	WHO	All
BERT (Bi-encoder)	44.57	64.24	81.12	50.55	81.85	64.47
BERT (Poly-encoder)	47.25	65.30	81.49	48.50	81.19	64.75
BART (Bi-encoder)	47.13	67.62	86.01	55.06	85.48	68.26
BART (Poly-encoder)	49.17	66.98	85.75	54.27	87.46	68.73

Table 9: The accuracy of different RQA models on the validation set. All of the results are averaged across 3 runs.

Methods	Australia	US	Canada	UK	WHO	All
BART RQA model						
BART RQA model	49.17	66.98	85.75	54.27	87.46	68.73
+ FEEDBACKRERANKER with explanation-based rating	51.34	69.09	84.20	56.87	87.79	69.86
+ FEEDBACKRERANKER with rating only	51.09	68.57	86.84	58.21	88.78	70.70
BERT RQA model						
BERT RQA model	47.25	65.30	81.49	48.50	81.19	64.75
+ FEEDBACKRERANKER with explanation-based rating	51.34	70.15	83.72	53.71	84.49	68.68
+ FEEDBACKRERANKER with rating only	51.09	68.46	84.18	55.69	85.15	68.91

Table 10: Accuracy of PIPELINE models using different feedback data to train the re-ranker on the validation set. All of the results are averaged across 3 runs.