When Life Gives You Lemons 🍋, Make Cherryade 🍒: *Converting Feedback from Bad Responses into Good Labels*

Weiyan Shi[†] Stanford University

Emily Dinan Meta AI

Kurt Shuster[◇] Character.ai

Jason Weston[*] Meta AI

Jing Xu[*] Meta AI

Abstract

Deployed dialogue agents have the potential to integrate human feedback to continuously improve themselves. However, humans may not always provide explicit signals when the chatbot makes mistakes during interactions. In this work, we propose JUICER, a framework to make use of both binary and free-form textual human feedback. It works by: (i) extending sparse binary feedback by training a *satisfaction classifier* to label the unlabeled data; and (ii) training a *reply corrector* to map the bad replies to good ones. We find that augmenting training with model-corrected replies improves the final dialogue model, and we can further improve performance by using both positive and negative replies through the recently proposed DIRECTOR model.

1 Introduction

Existing dialogue models are primarily trained on human-human conversations (Conneau et al., 2019; Baumgartner et al., 2020; Smith et al., 2020). As dialogue agents become increasingly powerful and carry substantial conversations with humans (Shuster et al., 2022b), it becomes pressing to have the models learn from dialogue successes and failures in the wild, and hence improve after deployment.

Prior work has studied how to collect and learn from feedback in human-model dialogues (Li et al., 2016a,b; Hancock et al., 2019; Xu et al., 2022). But most existing methods were proposed under settings where either feedback can be obtained whenever needed or all turns are annotated with human feedback. For instance, Xu et al. (2022) introduced a dataset with all turns annotated by crowdworkers with three types of feedback: (1) binary thumbs up/down; (2) free-form textual feedback on what went wrong; (3) gold corrections on what the bot should have said instead. Unfortunately, annotations such as thumb ups/downs and gold corrections are often sparse in real-life deployment settings. For example, human conversationalists give thumbs up/down to bot messages in conversations with the deployed BlenderBot3 model around 5-6% of the time (Shuster et al., 2022b). On the other hand, human conversationalists may express their dissatisfaction with bad responses and explain what went wrong more naturally in free-form textual feedback as part of the conversation, rather than providing the exact gold corrections to those bot responses. Therefore, in this paper we study how to utilize sparse binary and gold correction feedback, and relatively dense free-form textual feedback to improve dialogue models during deployment.

In this work, we introduce JUICER, a framework to "squeeze the juice" out of the sparse human feedback in human-model conversations to improve the dialogue models after deployment. JUICER consists of four steps: (1) we first train a binary *satisfaction classifier* and a *reply corrector* on existing binary feedback and gold corrections; (2) we then use the *satisfaction classifier* to label all the bot responses that are missing human labels; (3) next we use the *reply corrector* to correct bad bot responses (lemons 🍋) into good ones, conditioning on human textual feedback; (4) finally we augment the training data with the new good responses (cherryade 🍒) and re-train our final dialogue models.

To evaluate JUICER on state-of-the-art chatbots in such a setting, we thus construct a new sparse sampled version of the existing FITS dataset from Xu et al. (2022), which consists of fully annotated human-model conversations between users and existing state-of-the-art internet-augmented models such as BlenderBot 2 (Komeili et al., 2021; Xu et al., 2021) and SeeKeR (Shuster et al., 2022a).

We explore a variety of methods to take advantage of limited human feedback at each step of the JUICER framework. Our main results are:

† Work done when interning at Meta AI.
◇ Work done at Meta AI.
* Equal contribution.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 3066–3082

- We show that free-form textual feedback is a very useful signal for improving the performance of both a *satisfaction classifier* to identify good and bad responses, and a *reply corrector* to generate better corrections.
- Augmenting training data with *reply-corrector*-generated corrections works better than only training with existing gold corrections.
- Models such as DIRECTOR (Arora et al., 2022) that utilize both gold/predicted good and bad responses further improves the final dialogue model. Our final best models outperform the baseline BlenderBot 2 model or using DIRECTOR alone.

2 Related Work

Many recent works have studied how to align language models with human feedback (Nakano et al., 2021; Ouyang et al., 2022; Scheurer et al., 2022; Saunders et al., 2022; Schick et al., 2022). For instance, InstructGPT (Ouyang et al., 2022) was fine-tuned using feedback from labelers who ranked model outputs. Scheurer et al. (2022) fine-tuned GPT-3 and InstructGPT on 100 examples of free-form textual feedback from humans to improve summarization tasks and found that only the larger models such as GPT-3 (175B) (Brown et al., 2020) can generate accurate refinements using feedback. Saunders et al. (2022) fine-tuned large language models to generate self-critiques for summarization tasks to assist human annotators, and continued to refine the models on feedback. In this work, we focus on improving dialogue agents given various human feedback signals (binary, free-form natural language and gold corrections) and compare our methods to Scheurer et al. (2022).

Existing works have also studied how to correct language model output. For instance, Elgohary et al. (2021) proposed a model to understand natural language feedback and produce a series of edits to correct a text-to-SQL semantic parser. Tandon et al. (2022) trained a memory-augmented corrector to convert feedback to edits and fix model outputs for a script generation task. Some recent large language model research can also repair generations given human feedback (Scheurer et al., 2022; Saunders et al., 2022).

Past research has also explored how to integrate feedback into dialogue agents (Li et al., 2016a,b; Hancock et al., 2019; Shuster et al., 2020; Xu et al., 2022). Li et al. (2016a) investigated how to improve the chatbot's question-answering ability with general textual feedback in a reinforcement learning setting. Hancock et al. (2019) developed a self-feeding chatbot that can construct new examples from existing human-bot conversations and ask for feedback when necessary to improve itself. Xu et al. (2022) proposed a dataset with internet-augmented dialogues, where each turn is annotated with human feedback, and they found that continuously retraining the model on binary feedback after deployment is helpful. Our work focuses on converting bad responses into good ones to augment the data and learn from feedback about failures.

3 Human Feedback Setting

As illustrated in the dialogue example in Figure 1, we consider a deployed system where one can collect three types of feedback:

(1) **binary feedback**, where the human conversationalist explicitly likes () or dislikes () a bot response;

(2) **free-form textual feedback**, where the human explains conversationally what was wrong when they dislike a response (e.g., *"That's a quick topic change! Let's continue to talk about fruit, perhaps fruit drinks?"*);

(3) **gold correction**, where the human conversationalist suggests an alternative reply the bot should have said, (e.g., *"I like watermelons too! They tastes great in drinks."*).

In a deployment setting, it is unnatural to ask users to always click the thumbs up and down and write gold corrections whenever the bot makes a mistake. Instead, users tend to provide free-form textual feedback on what was wrong in their dialogue response to express dissatisfaction when the bot makes errors (See and Manning, 2021). Therefore many responses may be missing binary feedback (Shuster et al., 2022b). In this paper, we consider a sparse thumbs up/down signal and sparse gold correction signal setting, but a dense free-form textual feedback signal (i.e., mistakes are followed by textual feedback). After collecting conversations with these feedback signals, we can consider methods to utilize them to improve the dialogue model.

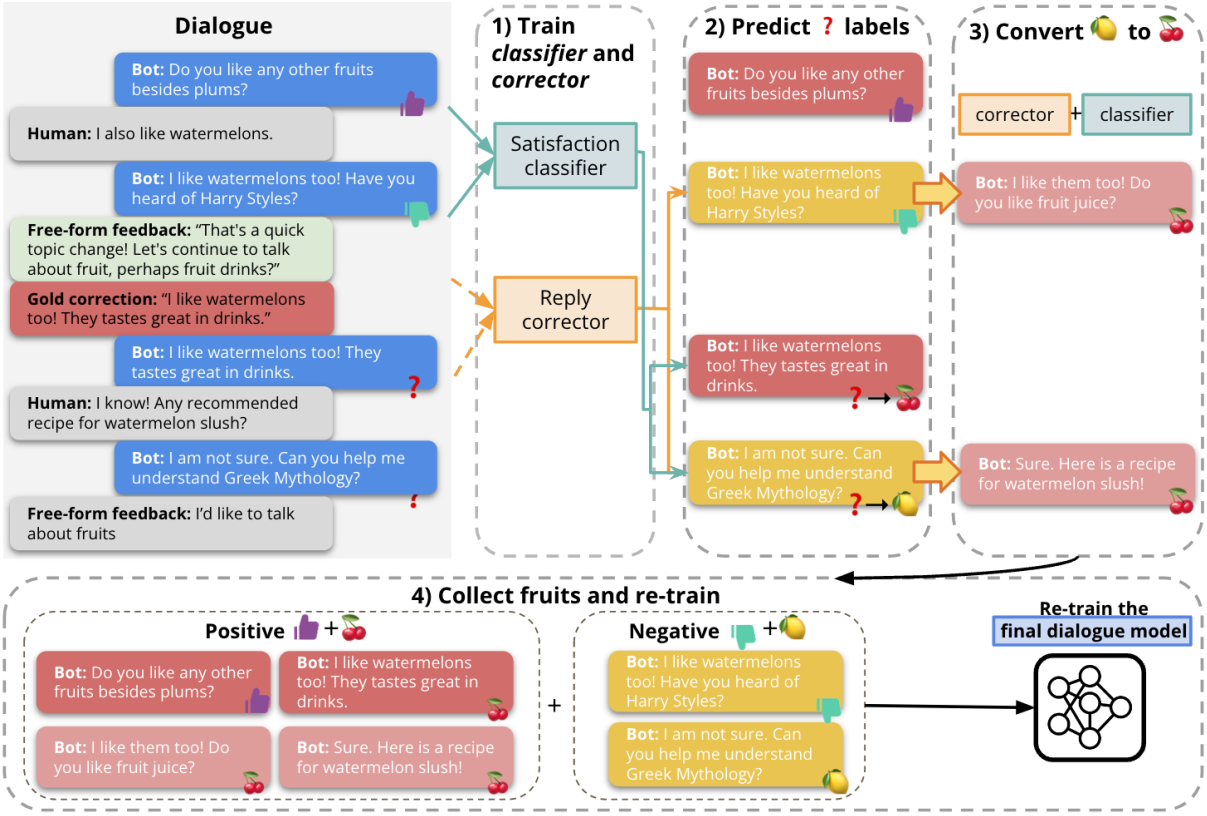

Figure 1: Our JUICER model. During deployment, we collect three types of human feedback: (1) binary thumbs up and down; (2) free-form textual feedback on what was wrong with the reply (*"That's a quick topic change! Let's continue to talk about fruit, perhaps fruit drinks?"*); (3) gold corrections of poor replies (*"I like watermelons too! They tastes great in drinks."*). In JUICER, (1) we first train a *satisfaction classifier* and a *reply corrector* on existing feedback, (2) we then use the *satisfaction classifier* to predict binary satisfaction labels for the un-annotated turns, (3) next we use the *reply corrector* to convert the bad replies to good ones, (4) finally we collect the good and bad replies including corrections and re-train the final dialogue model to improve it with human feedback.

4 The JUICER Method

Figure 1 shows the overview of our framework JUICER to incorporate limited human feedback to improve the deployed dialogue model. The framework consists of training a *satisfaction classifier*, a *reply corrector*, and eventually the final dialogue model itself. We define the notation here. For a given bot reply: (1) ? denotes un-annotated turns; (2) 👍 and 👎: annotated as good or bad responses by users, as defined before; (3) 🍋: predicted as bad by the *satisfaction classifier*; (4) 🍒: predicted as good by the *satisfaction classifier*.

JUICER involves four steps, summarized here:

1. ***Step 1.*** Train two supervised models: a *satisfaction classifier* to detect good and bad replies, and a *reply corrector* to correct the bad replies to good ones.

2. ***Step 2.*** Apply the *satisfaction classifier* to predict binary labels for all replies missing binary feedback (? → 🍒 or 🍋). After this step, each bot reply has a label.

3. ***Step 3.*** Use the *reply corrector* to convert the bad replies that are either disliked by human users or are predicted as bad by the *satisfaction classifier* in *Step 2* to good replies (👎 → 🍒 , 🍋 → 🍒).

4. ***Step 4.*** Re-train the final dialogue model by augmenting the training data with the good (👍 + 🍒) and bad (👎 + 🍋) replies derived from human feedback and the predictions from the previous steps.

Now we describe each step in more details.

4.1 Step 1: Train *satisfaction classifier* and *reply corrector* on existing feedback

We first train two models: (1) a *satisfaction classifier*, and (2) a *reply corrector* in order to build an

augmented training set in later steps. In our experiments, both models are trained with human-labeled data which come from the FITS task (Xu et al., 2022), described further in Section 5.1.1.

(1a) *Satisfaction classifier* The training target of the *satisfaction classifier* is a binary satisfaction label (👍 or 👎). For the input to the classifier, we experimented with two variants: (1) the context + the bot reply to be labeled, and (2) the context + the bot reply to be labeled + the next human response. As shown in the example in Figure 1, when the first bot reply is given a thumbs-up, the next human response is a natural continuation of the conversation (e.g., *"I also like watermelons"*); when the bot reply is disliked (the second bot reply), the next human response is free-form textual feedback on what went wrong (e.g., *"That's a quick topic change! Let's continue to talk about fruit, perhaps fruit drinks?"*). Hence, the next human response can be indicative of the quality of its preceding bot reply, and we include it in the input. In our experiments, the *satisfaction classifier* is trained by fine-tuning a 311M-parameter transformer pre-trained on pushshift.io Reddit data (Baumgartner et al., 2020).

(1b) *Reply corrector* The input to the *reply corrector* is the context + the bad bot reply to correct + the next human free-form textual feedback on what went wrong. The training target is the correction to the bad reply which can be either (1) gold corrections written by crowdworkers; or (2) the next bot replies from the original FITS data that are classified as good ("self-corrections"). We fine-tuned the *reply corrector* from a 3B parameter R2C2 transformer model (Shuster et al., 2022a).

4.2 Step 2: Predict missing labels

In a bot-human dialogue, the binary feedback can be quite sparse, with many replies having no explicit feedback. We thus predict labels for these replies with the *satisfaction classifier* trained in Step 1a. After this step, every bot reply in the dataset has a binary label either from the original human binary feedback (👍 or 👎), or predicted by the *satisfaction classifier* (🍒 or 🍋).

4.3 Step 3: Convert lemons to cherries

We can now augment the training data. We use the *reply corrector* trained in step 1b to generate improved replies for any examples labeled as bad (👎 or 🍋), and then add them to the training set for the final dialogue model.

Selecting correctable cases However, we note that not all bad responses are easily correctable given free-form textual feedback. For example, the human feedback "You are talking nonsense!" could help indicate this is a 🍋 using the *satisfaction classifier*, but is less helpful for knowing what the right response is, compared to more constructive feedback such as "That's a quick topic change! Let's continue to talk about fruit, perhaps fruit drinks?" We thus experiment with detecting cases that are "correctable", and only use these to augment our training data. We first embed the free-form textual feedback and the immediate next bot reply in recorded conversations with Sentence-BERT (Reimers and Gurevych, 2019), and then calculate their cosine similarity score. If the score is high, it means that the human free-form textual feedback is easier for a model to comprehend and thus revise its own response accordingly. We define such examples as correctable and then threshold the similarity score to pick out correctable cases.

Predicting reply corrections To obtain the corrections, we adopt a reranking-based learning method widely used in many previous studies (Nie et al., 2020; Nakano et al., 2021; Askell et al., 2021) to score and rank the generations. We first use the *reply corrector* to generate many correction candidates (60 in our experiments). Then we concatenate the original context with the correction candidates and feed them into the *satisfaction classifier* from Step 1a. Finally, we select the top one with the highest probability output by the classifier as the final correction. If all generated corrections are predicted as bad, we will skip this example.

4.4 Step 4: Collect fruits and re-train

After the previous steps, each bot response is annotated with either a gold or predicted binary label, and those labeled as bad are converted from bad responses to good ones using human feedback. The final step is to augment the training set of the final dialogue model with the new data.

One straightforward method to improve the model is to augment the training data with all the positive replies including the corrections (👍 + 🍒) and use the standard language modeling objective. However, this standard training does not utilize negative/bad replies (👎 + 🍋) to avoid them. We

hence also apply the recently proposed DIRECTOR model (Arora et al., 2022) to both reinforce the positive responses and penalize the negative ones. DIRECTOR is a unified decoder-classifier model jointly trained with a language modeling task and a classification task. During inference, it uses its language modeling head to predict the next token probability, and its classifier head to decide if the tokens belong to positive examples to generate the final output. But it is worth noting that in this step in JUICER, we could use any other approach that utilizes both positive and negative responses to retrain and improve the final dialogue model.

5 Experimental Setup

In our experiments, we used the 3B parameter BlenderBot2 (BB2 3B) (Komeili et al., 2021; Xu et al., 2021) as the base dialogue model and try to improve it with human feedback from deployment.

5.1 Datasets: FITS and DEMO

We performed experiments on the FITS (Xu et al., 2022) dataset. We also tested the zero-shot transferability of both the *satisfaction classifiers* and the *reply correctors* on a real deployment dataset DEMO (Ju et al., 2022).

5.1.1 FITS

FITS contains internet-augmented human-bot dialogues with annotated feedback for every turn, including a binary label, free-form textual feedback and a gold response, with around 39k bot utterances in total. See Section A.1 for more details. To mimic a deployment setting with limited feedback, we uniformly sampled 20% of the bot responses from the training set of FITS and considered them to have binary feedback and gold labels, while the rest were considered unlabeled. However, we did not remove free-form textual feedback when it is present, as it remains part of the conversation, see Figure 1. Table 4 in the Appendix shows the data statistics after sampling.

We used the original FITS validation, test set and unseen test set (of new conversational topics) for evaluation, and employed the same metrics as Xu et al. (2022) for the final dialogue models: perplexity, F1 overlap with the gold annotation, and human evaluation via conversations with the bot. During conversations, crowdworkers click 👍 or 👎 per turn and give a final rating (a score out of 5) in the end. We report the average good response rate in percentage. See Appendix C for more details

5.1.2 DEMO

The dataset DEMO is from the deployment of BlenderBot 3 (Shuster et al., 2022b) with responses verified by crowdworkers (Ju et al., 2022). In total 923 bot responses across 81 conversations are used as an evaluation set.

5.2 Baselines

We have two categories of baselines: (1) without model-augmented data, and (2) with a prompt-based *reply corrector*. In addition, we also compare with oracle methods using 100% labeled feedback data without sampling.

Baselines without augmentation. The most straightforward baselines are to fine-tune with the limited human-labeled feedback only.

- **Gold corrections from 20%** Gold corrections provide a strong learning signal. Here, we simply fine-tune BB2 3B on the gold corrections from 20% human-annotated set.
- **Free-form textual feedback from 20%** Following Hancock et al. (2019), we fine-tune BB2 3B with the context as the input and the free-form textual feedback (identified as the response following the bad 👎 responses) as the target.

Baseline with a prompt-based *reply corrector*. Instead of training a supervised *reply corrector* with gold corrections, this baseline prompts an off-the-shelf model with free-form textual feedback and instructions like "given the feedback, correct the original response" as a *reply corrector* to generate corrections, and then fine-tunes the final dialogue model on these corrections.

- **3B-all-corrections**: Scheurer et al. (2022) proposed an approach to improve language models with language feedback, originally applied to summarization tasks, which we adapt here for dialogue. Given a small number (100) human feedback samples, they prompted a language model to condition on the context (input+feedback) to re-generate multiple summarization corrections, picked the correction with the highest similarity score with the feedback, and finally fine-tuned the language model on the corrections to improve it. In our implementation, we use the baseline BlenderBot 2 model (3B) as the *reply*

corrector. While Scheurer et al. (2022) used larger language models (175B), our implementation of the baseline is more comparable to our JUICER models since our *reply corrector* also has 3B parameters. In our experiments, instead of using only 100 examples, we make this a stronger baseline by generating corrections for *all* the bad replies.

5.3 JUICER models

We also compare several variants of JUICER.

- **JUICER**. We fine-tune BB2 3B by augmenting the 20% human-annotated data with (1) predicted good responses by the *satisfaction classifier* from the remaining 80% unannotated turns, and (2) predicted corrections generated by the *reply corrector*, filtered to only include the correctable cases rather than using all the predicted corrections.
- **JUICER + DIRECTOR**. We fine-tune using DIRECTOR which uses both the positive and negative replies. Both gold annotations and the filtered corrections generated by the *reply corrector* are used as positive classification data. Bad responses labeled by humans or the *satisfaction classifier* are used as negative data for fine-tuning the classifier head.
- **w/o predicted corrections (from Step 3)**. In this ablation, we fine-tune the final dialogue model with only predicted good responses by the *satisfaction classifier*, without the corrections generated by the *reply corrector*.
- **w/o selecting correctable cases**. In this ablation, we only augment with (1) predicted good responses by the *satisfaction classifier*, and (2) all the predicted corrections without selecting the more correctable cases. This tests if selecting correctable cases brings improvements.

6 Results

We first evaluate the *satisfaction classifier* (Table 1a), and the *reply corrector* (Table 1b). We then perform both automatic and human evaluations on the final dialogue models (Table 2 and Table 3).

6.1 *Satisfaction classifier*

Table 1a shows the classifiers' performance on the FITS data and also their zero-shot performance on DEMO.

Adding the next human response helps. We find the balanced accuracy of detecting satisfaction using only the dialogue context and the bot response itself is ∼75% on FITS. It is significantly improved to ∼95% by including the next human message in the input. A similar improvement is found when measuring balanced F1 as well. On the deployment dataset DEMO where organic users are not required to always write free-form textual feedback when seeing a bad reply, adding the human response still improves the balanced F1 from 64.77 to 71.24, despite this being zero-shot performance (without training on this dataset). These results indicate the importance of using the next human message to make satisfaction classification decisions. As using the next human response helps, we default to using this *satisfaction classifier* variant in our standard JUICER setup.

6.2 *Reply corrector*

Table 1b shows the results of training the *reply corrector*, comparing different input feature choices.

Free-form textual feedback improves the correction. We performed an ablation study where the *reply corrector* trains on (context + bad reply → *good reply*) without the free-form textual feedback, shown in "w/o free-form textual feedback". As expected, adding free-form textual feedback on what went wrong improves the *reply corrector*'s performance. The best results are relatively close to the oracle performance which uses 100% (rather than 20%) gold data for training (23.39 F1 vs. 21.41 and 2.93 PPL vs. 3.07).

Augmenting with self-correction pairs helps. The standard *reply corrector* trains on "gold-correction" pairs (context + bad reply + free-form textual feedback→ gold correction). Besides these human-written gold corrections, we can also train the *reply corrector* on "self-correction" pairs (context + bad reply + free-form textual feedback→ *good bot reply*), where a bad reply is followed by a good bot reply either liked by humans or predicted as good by the *satisfaction classifier*, suggesting that the bot "corrects" itself in the following turn. We found that augmenting with these "self-corrections" improves the F1 from 17.10 to 21.41. We can also multitask with various dialogue tasks to further improve the *reply corrector*'s performance. See Section A.4.1 for more details.

(1a) *Satisfaction Classifier* Input	Valid Acc↑	Valid F1↑	Test Acc↑	Test F1↑	Test Unseen Acc↑	Test Unseen F1↑	DEMO (zero-shot) Acc↑	DEMO (zero-shot) F1↑
context+bot+human	**94.66**	**97.25**	**95.76**	**97.83**	**96.74**	**98.34**	**59.73**	**71.24**
context+bot	75.58	86.07	74.53	85.38	71.46	83.25	56.60	64.77

(a) *Satisfaction classifier* results (classification balanced accuracy and balanced f1) on both FITS and DEMO (zero-shot). Adding the next human message helps the satisfaction prediction, even in the zero-shot case.

(1b) *Reply Corrector* Input	Valid F1↑	Valid PPL↓	Test F1↑	Test PPL↓	Test Unseen F1↑	Test Unseen PPL↓
Oracle with 100% annotations						
gold corrections from 100%	23.39	2.93	21.83	2.63	22.27	4.56
w/ free-form textual feedback						
gold corrections from 20% + self-corrections	**21.41**	**3.07**	**20.20**	**2.75**	**21.77**	**4.66**
gold corrections from 20%	17.10	3.37	16.21	2.98	17.91	4.97
w/o free-form textual feedback						
gold corrections from 20% + self-corrections	18.80	3.13	18.36	2.82	18.97	4.84
gold corrections from 20%	16.41	3.40	15.08	3.04	16.46	5.06

(b) *Reply corrector* perplexity and F1 on valid/test/test unseen sets. Augmenting with self-corrections improves the result, comparable to the oracle model using 100% gold corrections. Using free-form textual feedback is helpful.

Table 1: Performance of the modules in Step 1: (a) *satisfaction classifier*, and (b) *reply corrector*.

Qualitative results show the corrections make sense. We also include generated correction examples on the FITS data in Appendix Table 8 and on the deployment data in a zero-shot fashion in Appendix Table 9. These examples show that the *reply corrector* can integrate free-form textual feedback to correct the bad replies, even for zero-shot deployment data.

See section A.4.2 for further details and results on the *reply corrector* evaluation.

6.3 Final dialogue model evaluations

The final dialogue model results are given in Table 2 (automatic evaluations) and Table 3 (human evaluations). All methods are fine-tuned from the 3B parameter BlenderBot 2 (BB2), making the models comparable.

Using JUICER to augment data improves results. JUICER yields significant gains over the baseline transformer BB2 3B in both automatic evaluations and human evaluations. For example, we see an F1 increase from 15.3 to 18.5 on the unseen test set, and an improvement of good responses from 33.2% to 41.9% in human evaluations. JUICER also performs better than baselines without augmentation (e.g., gold corrections from 20%).

Our supervised *reply corrector* outperforms a prompt-based one. Compared to the prompt-based *reply corrector* baseline (Scheurer et al., 2022), all the JUICER models perform better in automatic evaluations. When the prompt-based model is used as a *reply corrector* to produce corrections to augment the final dialogue model training, the final model evaluation (F1=14.2, ppl=8.9) is worse than augmenting with the corrections in JUICER (F1=16.7, ppl=8.5).

Augmenting training with predicted corrections in JUICER helps. JUICER augments training with predicted corrections, which improves both the F1 and perplexity across the board compared to JUICER without predicted corrections, e.g. 18.5 vs. 17.9 on the test unseen F1. This makes sense because the predicted corrections are generated by the *reply corrector* given human free-form textual feedback which contains valuable information, and fine-tuning the final dialogue model on these corrections can steer it toward better replies.

Selecting correctable cases can help. JUICER picks only correctable cases to augment the training data, with around 62% of cases selected (threshold chosen based on the validation set). Compared to naively augmenting with all predicted corrections, we see gains on valid and unseen test F1 (18.5 vs. 18.0), although there is no gain on the seen test set.

Final dialogue model	Automatic evaluation					
	Valid		Test		Test Unseen	
	F1↑	PPL↓	F1↑	PPL↓	F1↑	PPL↓
BB2 3B	14.4	10.6	14.7	10.3	15.3	9.3
+gold corrections from 20%	16.2	9.1	15.6	8.9	17.9	8.4
+free-form textual feedback from 20%	13.1	10.4	12.6	10.3	13.7	9.6
3B-all-corrections (prompt-based)	14.2	8.9	14.5	8.7	15.2	8.2
JUICER models						
+JUICER	16.7	**8.5**	16.2	8.4	**18.5**	**8.0**
+JUICER +DIRECTOR	**17.2**	n/a	**16.7**	n/a	17.7	n/a
JUICER ablations						
w/o predicted corrections	15.7	9.0	15.8	8.8	17.9	8.2
w/o selecting correctable cases	16.4	8.5	16.4	8.4	18.0	8.1

Table 2: Final dialogue model automatic evaluation results. All the dialogue models are fine-tuned from BB2 3B. JUICER models with augmentations are better than the baselines without augmentations. JUICER with a supervised *reply corrector* also performs better than the baseline with a prompted-based *reply corrector*. DIRECTOR utilizing negative examples is effective. Using predicted corrections and selecting correctable cases are useful.

Final dialogue model	Human evaluation	
	Good% ↑	Rating ↑
BB2 3B	33.2%	3.09
+gold corrections from 20%	**39.4%**	2.89
JUICER models		
+JUICER	**41.9%**	3.06
+JUICER +DIRECTOR	**45.5%**	**3.34**

Table 3: Final dialogue model human evaluation results. We report the % of good responses and the overall rating, as judged by crowdworkers during conversations. We bold statistically significant improvements (independent two-sample t-test, $p < 0.05$) of methods over the BB2 3B baseline. JUICER outperforms the baselines. JUICER +DIRECTOR works the best.

DIRECTOR provides further gains. DIRECTOR utilizes both the (predicted) binary feedback signal and textual feedback signal to penalize negative responses. Applying it improves the results further over standard JUICER (45.5% good responses vs. 41.9% for JUICER *without* DIRECTOR, as measured by human evaluations). Because DIRECTOR uses a classifier head to decide if a token should be included in the final generation, the distribution is altered and perplexity measures are not applicable. However, it gives gains in F1 on valid and test sets, although not on the unseen test set. JUICER and DIRECTOR together also outperforms DIRECTOR alone, even when DIRECTOR uses 100% gold binary labels, see Appendix Table 11. Further variants and experiments with DIRECTOR are also given in Section A.5.2.

JUICER achieves comparable results to methods with oracle access to gold labels. Compared to methods using 100% gold data which was not given to JUICER, our best JUICER models achieve comparable performance, especially on F1 and human evaluation. For example, test unseen F1=17.6 for the best "oracle" method vs. 17.7 for the best JUICER model, 47.0% vs. 45.5% good responses, and 3.38 vs. 3.34 in human ratings. See Appendix Table 11 for further details. These "100% data" methods can be seen as upper bound results, showing that JUICER does extract most of the signal from the portion of the dialogue data without binary or gold feedback.

See Section A.5 for further experiments and details on final model evaluations.

7 Conclusion

Deployed dialogue agents should continuously improve by using human feedback gathered during interactions. Unfortunately, feedback collected in the wild can be limited. In this paper, we proposed JUICER, a framework to efficiently use limited organic feedback signals (binary labels and gold corrections) if free-form textual feedback is provided. JUICER works by correcting bad responses into good ones to augment the training data for the final dialogue model. Experiments show that augmenting with such predictions can integrate human feedback and improve overall performance.

8 Limitations and Discussions

In our experimental setting, we assume dense free-form textual feedback, i.e., a bad reply is always followed by a free-form message explaining what was wrong. In real deployments, this free-form textual feedback signal may not always be given and without it, the binary *satisfaction classifier* may not necessarily achieve a high accuracy or F1 (e.g., 90+), which could also impact the later steps. It remains to be seen in real deployments how dense this signal is, and what methods can be used to encourage users to make these signals as dense as possible, so that strong feedback signals are available to train on.

We have also assumed good intent from human conversationalists, but it is possible to have adversarial and bad actors interacting with the bot. In particular, incorrect feedback or opposite feedback (e.g., thumbs up instead of down) could be supplied by the human for incorrect bot behavior. We see this as an important research direction that should be pursued in parallel to work on algorithms like the ones we study here. See e.g. Ju et al. (2022) for recent work addressing bad actors and adversarial feedback.

The training/evaluation loop of JUICER can be long due to its iterative nature. The advantage of using a *reply corrector* is that we can qualitatively evaluate the quality of the generated corrections. But the drawback is that we need to first train a *reply corrector*, use it to generate corrections, and finally improve the dialogue. We assume that the best *reply corrector* will lead to the best final dialogue model, but this remains to be studied. Another possible direction is to use a latent *reply corrector* to integrate the feedback in a more end-to-end fashion instead of a supervised *reply corrector* that will generate explicit corrections separate from the dialogue model.

Additionally, the proposed JUICER framework improves the dialogue model offline rather than correcting the response on the fly. With the necessary infrastructure support, there is potential for improving the models online. This could be a natural setting for reinforcement learning to get interactive feedback and iteratively update the model policy as the conversation continues. Such a direction does not come without dangers, however, such as the model degrading if it receives poor inputs, e.g. from bad actors as mentioned before.

References

Kushal Arora, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. 2022. Director: Generator-classifiers for supervise language modeling. *arXiv preprint arXiv:*.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. *arXiv preprint arXiv:2001.08435*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Ahmed Elgohary, Christopher Meek, Matthew Richardson, Adam Fourney, Gonzalo Ramos, and Ahmed Hassan Awadallah. 2021. Nl-edit: Correcting semantic parse errors through natural language interaction. *ACL*.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415*.

Da Ju, Jing Xu, Y-Lan Boureau, and Jason Weston. 2022. Learning from data in the mixed adversarial non-adversarial case: Finding the helpers and ignoring the trolls. *arXiv preprint arXiv*.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*.

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

Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2016b. Learning through dialogue interactions by asking questions. *arXiv preprint arXiv:1612.04936*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders,

et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Yixin Nie, Mary Williamson, Mohit Bansal, Douwe Kiela, and Jason Weston. 2020. I like fish, especially dolphins: Addressing contradictions in dialogue modeling. *arXiv preprint arXiv:2012.13391*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*.

Jérémy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2022. Training language models with language feedback. *The First Workshop on Learning with Natural Language Supervision at ACL 2022*.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2022. Peer: A collaborative language model. *arXiv preprint arXiv:2208.11663*.

Abigail See and Christopher D Manning. 2021. Understanding and predicting user dissatisfaction in a neural generative chatbot. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–12.

Kurt Shuster, Mojtaba Komeili, Leonard Adolphs, Stephen Roller, Arthur Szlam, and Jason Weston. 2022a. Language models that seek for knowledge: Modular search & generation for dialogue and prompt completion. *arXiv preprint arXiv:2203.13224*.

Kurt Shuster, Jack Urbanek, Emily Dinan, Arthur Szlam, and Jason Weston. 2020. Deploying lifelong open-domain dialogue learning. *arXiv preprint arXiv:2008.08076*.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. 2022b. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*.

Eric Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. ACL.

Niket Tandon, Aman Madaan, Peter Clark, and Yiming Yang. 2022. Learning to repair: Repairing model output errors after deployment using a dynamic memory of feedback. *NAACL Findings.(to appear)*.

Jing Xu, Arthur Szlam, and Jason Weston. 2021. Beyond goldfish memory: Long-term open-domain conversation. *arXiv preprint arXiv:2107.07567*.

Jing Xu, Megan Ung, Mojtaba Komeili, Kushal Arora, Y-Lan Boureau, and Jason Weston. 2022. Learning new skills after deployment: Improving open-domain internet-driven dialogue with human feedback. *arXiv preprint arXiv:2208.03270*.

A Appendix

A.1 Details on FITS

In this section, we describe the existing FITS task (Xu et al., 2022) in more details. In FITS, each bot message is annotated with the following feedback. The setting also ensures that after the human provides the feedback, the conversation can be continued with the feedback integrated.

- A binary satisfaction label.
- If it is a bad reply, the human provides free-form textual feedback on what went wrong in the next human message.
- Multiple-choice selection on what the bot could do to improve this turn:
  (a) using a better search query; or
  (b) attending better to the search results; or
  (c) other issues with the overall reply; or
  (d) no issue (a good reply).
- If selecting (a), the human provides a better search query, which will be used in the next turn to continue the conversation.
- If selecting (b), the human is presented with the search results and selects the relevant sentences, which will be added to the model input in the next turn.
- If selecting (c), the human provides a better overall reply (a gold correction), which is copied to be the next bot response.

A.2 Sampling FITS

In our experiments, we uniformly sample 20% of the FITS training set to mimic a deployment setting with sparse binary and gold feedback. Table 4 shows the sampled dataset statistics. Out of the 20% of the training set with labels, 1376 examples are "better overall reply" annotated with gold corrections, which accounts for 7% of all bad responses to be corrected in FITS. Those 1376 corrections will later be used to train the *reply corrector* for augmenting limited human feedback. The size of 20% of the FITS training set (7768 examples) is also similar to that of the validation and test sets.

Feedback Breakdown	Train (20%)	Valid	Test	Test Unseen
Total	7768	4245	9726	8907
Better Search Query	1056	605	1167	1036
Better Results Usage	1383	756	1527	1310
Better Overall Reply	1376	714	1493	1372
Good Response	3953	2170	5539	5189

Table 4: Data statistics of the sampled version of FITS used in our experiments. We sampled 20% from FITS. Note that the training set size of labeled binary feedback is similar to the test sets.

A.2.1 Varying the sampling rate

As an ablation study, we varied the sampling rate. Table 5 shows different final dialogue models' results with different sampling rates. The input to the dialogue model is the context, and the output is the human-written gold correction.

As we increase the sampling rate, the final dialogue model's perplexity improves in general, but the gain becomes smaller. For instance, when we only sample 5% from FITS, the validation perplexity is 9.52; if we increase the sampling rate to 20%, the perplexity is 9.09; but when we further increase the sampling rate to 30% and 50%, the perplexity becomes 9.12 and 8.80 respectively.

We find the 20% sampling rate is a good balancing point with both reasonable F1 and perplexity.

A.3 Different modules in JUICER

There are different modules involved in JUICER and we summarize them in Table 6. To sum up, JUICER has two helper modules, a *satisfaction classifier* and a *reply corrector* to help improve the final dialogue model.

The *satisfaction classifier* identifies if the bot response is satisfactory or not. It is evaluated on both FITS and zero-shot DEMO deployment datasets.

Both the *reply corrector* and the final dialogue models are generative models, and are automatically evaluated on the human-written gold corrections in the FITS validation and test sets, as gold corrections can reflect the model's ability to generate good responses.

The *reply corrector* converts bad responses into good ones using free-form textual feedback. We evaluate it on gold search results instead of live search results, in order to generate better reply corrections. We describe it in more detail in Section A.4.

The final dialogue model is evaluated on live search results from Bing, filtered by Common-

Final dialogue model Varying the sampling rate	Valid		Test		Test Unseen	
	F1↑	PPL↓	F1↑	PPL↓	F1↑	PPL↓
Baseline performance varying the sampling rate						
+gold correction from 20%	16.2	9.1	15.6	8.9	17.9	8.4
+gold correction from 5%	16.8	9.5	16.6	9.3	18.9	8.5
+gold correction from 10%	16.6	9.3	16.5	9.1	18.9	8.4
+gold correction from 30%	15.7	9.1	16.3	8.6	18.1	8.3
+gold correction from 50%	16.5	8.8	16.0	9.0	17.9	8.3

Table 5: Final dialogue model results varying the sampling rate. Perplexities get better as we increase the sampling rate, but the gain becomes smaller. F1 first gets worse and then goes up. These suggest that a sampling rate of 20% is a good balancing point with both a good F1 and a good perplexity.

Model	Inputs → Outputs	Fine-tuned from	Evaluated on	Description
(1a) *Satisfaction classifier*	Context + bot reply (+ the next human response) → binary {good, bad} on the bot reply	311M Transformer	FITS valid&test and DEMO	Given the context, a bot reply and potentially the next human message, detect if the bot reply is satisfactory
(1b) *Reply corrector*	Context + bad bot reply + free-form textual feedback → improved reply ("a correction")	3B R2C2 (Shuster et al., 2022a)	Gold corrections in FITS valid&test (on gold search results)	Given the context, the bad reply, and free-form textual feedback, generate an improved reply ("correct the bad reply")
(2) Final dialogue model	Context → reply	3B BlenderBot 2	Gold corrections in FITS valid&test (on live search results)	Given the context, generate a reply. We fine-tune our models using BlenderBot 2 as the base model (Komeili et al., 2021; Xu et al., 2021).

Table 6: The input, output, and description of the three models used in JUICER.

Crawl, following Xu et al. (2022) instead of gold search results to better reflect performance with live users. We describe it in more detail in Section A.5.

A.4 *Reply corrector*

The *reply corrector* trains on data where for a given example the input consists of the dialogue context + bad reply to correct + the following human message, and the output consists of the correction for the bad reply. The models used in the main paper were fine-tuned from the R2C2 transformer (Shuster et al., 2022a).

A.4.1 Training the *reply corrector* with multiple tasks

In our experiments, we multi-tasked with various dialogue tasks to train the *reply corrector*, which improves the result. These tasks include the original reply correction task, the task with context as the input and free-form textual feedback as the target, and the dialogue task of Wizard of Internet (Komeili et al., 2021). We tuned the weights for different tasks, and other hyper-parameters (learning rate, batch size, etc) according to the performance on the validation set.

A.4.2 Evaluating the *reply corrector*

Since the *reply correctors* are used to generate corrections rather than interacting with live users, we evaluated them with gold search results (which leads to better corrections) instead of live search results from Bing (which better reflects the live interaction performance).

Although the *reply corrector* (Table 1b) and the final dialogue models (Table 2) are evaluated on the same validation and test subsets that have gold corrections, their results are not comparable because of the following two reasons. First, as mentioned earlier, the *reply correctors* condition on the gold search results instead of the live search results, while the final dialogue models use the live search results. Second, the *reply correctors* rely on the free-form textual feedback to convert lemons to cherries, so we also append the free-form textual feedback into the input to the *reply correctors*, but for the final dialogue model, we do not have the additional free-form textual feedback information. These are the main reasons why the results in Table 1b are better than those in Table 2.

A.4.3 Generating reply corrections

We adopt a reranking-based learning method to first generate multiple reply corrections, and then use the *satisfaction classifier* to score and rerank the generated corrections. Because the *reply corrector*'s performance is good (comparable to the one trained on 100% data in Table 1b) and we gen-

(1b) *Reply Corrector* Input	Valid F1↑	Valid PPL↓	Test F1↑	Test PPL↓	Test Unseen F1↑	Test Unseen PPL↓
fine-tuned from R2C2						
gold corrections from 20% + self-corrections	**21.41**	**3.07**	**20.20**	2.75	**21.77**	**4.66**
+ DIRECTOR	22.81	-	**22.59**	-	22.10	-
+ DIRECTOR OVERLAP	**23.00**	-	22.50	-	**22.55**	-
fine-tuned from BB2						
gold corrections from 20% + self-corrections	16.32	7.06	14.53	7.01	15.63	7.16

Table 7: *Reply corrector* results. The top block shows the *reply corrector* fine-tuned from R2C2 with DIRECTOR and DIRECTOR OVERLAP, and the bottom block shows the *reply corrector* fine-tuned from BB2. R2C2 is better than BB2 as a *reply corrector*. Using DIRECTOR improves the result. Using DIRECTOR OVERLAP further improves over DIRECTOR.

erated 60 correction candidates to choose from, the majority (99.96%, 16893 out of 16989) of bad responses have at least one generated correction that is predicted as satisfactory by the *satisfaction classifier*.

Generated reply correction examples. Table 8 and Table 9 show generated reply correction examples on the FITS dataset and the deployment dataset (zero-shot) respectively. These qualitative examples show that the *reply corrector* can convert bad replies into good ones using free-form textual feedback, even for unseen deployment data.

A.4.4 Using BB2 to train the *reply corrector*

The models used in the main experiments were fine-tuned from R2C2 (Shuster et al., 2022a). We also report results fine-tuned with BB2 in Table 7. We find that BB2 is worse than R2C2 as a *reply corrector* because its generated corrections are more like conversational replies rather than actual corrections.

A.4.5 Using DIRECTOR in the *reply corrector*

Using DIRECTOR to combine multiple feedback signals is also effective for the *reply corrector*. We can use DIRECTOR to further improve the *reply corrector*'s F1 to 22.81, as shown in Table 7, where the positive examples are the gold corrections and the negative examples are the bad bot responses. However, although the F1 of the DIRECTOR-enhanced *reply corrector* is better, we find that if we use it to generate reply corrections to improve the final dialogue models, the F1 is slightly better but the perplexity gets worse than using a regular *reply corrector* without DIRECTOR, as shown in Table 10. More analysis is needed to understand the reasons for this.

A.5 Final dialogue model evaluation

We evaluate the final dialogue model on live search results instead of gold search results to better reflect performance with live users.

A.5.1 Oracle performance using 100% feedback data

Xu et al. (2022) trained various methods on the entire FITS dataset. Since our method is trained only on 20% of FITS, the 100% models' performance could be viewed as an upper bound of our models. They also used the 3B parameter BlenderBot 2 as a base model for the final dialogue model, making it comparable to our experiments. Their results are in Table 11 and we detail their models below.

- **100% gold correction**. The input is the context and the target is the gold correction (6,601 in the entire FITS dataset). This can be directly compared to "gold correction from 20%" in Table 2.
- **100% free-form textual feedback**. The input is the context and the target is the free-form textual feedback. This should be compared to "free-form textual feedback from 20%" in Table 2.
- **100% module supervision**. BlenderBot 2 is an internet-augmented bot with different modules such as a search module to generate a search query, and a knowledge module to attend to the search results. Using the human-written gold search query, human-selected search doc and gold correction, they fine-tuned each individual module to improve BlenderBot 2.

From	Utterance
USR	Hi bot, what supplies do I need for a diaper change?
BOT	That's a good question. I'm not sure, but I'm sure you can find out online.
Feedback	*The results list at least one necessary supply, with the implication that bot would find others if it read further.*
Correction	Bring your supplies to the diapering area. Clean cloth diapers are essential. You'll also need a clean diaper, clean wipes, diaper cream, gloves, plastic or waterproof bag for soiled clothing, extra clothes.
Original	It depends on how many diapers you have. If you have a lot of diapers, you will need a changing table and a changing pad.

From	Utterance
USR	What is the name of hello kitty's sister?
BOT	Hello kitty's twin sister, mimmy.
Feedback	*That's great. I think to make this even better it could be phrased a little more conversationally. Something like "hello kitty has a twin sister, mimmy."*
Correction	Hello kittys sister is called mimmy! I think it's funny she looks a lot like her. Do you like hello kitties sister?
Original	Hello kitty has a twin sister named mimmy.

From	Utterance
USR	Can you tell me what two sides fought during the war of the roses?
BOT	The war of the roses was between the english and the french. The english won.
Feedback	*It looks like that answer is not right. The search results say something different.*
Correction	The war was fought between two competing royal families: the yorks and the lancasters, both members of the age-old royal plantagenet family.
Original	The english and the french.

From	Utterance
USR	What are risk factors of asthma?
BOT	Asthma is a chronic disease of the air passages in the lungs.
Feedback	*Try using a more specific query*
Correction	Asthma can be caused by a combination of genetic and environmental factors. These factors include: early exposure to allergens, changing in weather, respiratory infections and more.
Original	Asthma is a chronic disease of the air passages in the lungs.

Table 8: Qualitative correction examples generated by the best *reply corrector* on FITS.

- **100% reward-based learning**. They also adopted a reward-based learning approach, and built a *satisfaction classifier* to identify good and bad responses. They used the vanilla BlenderBot 2 model to generate multiple responses, then reranked them with the score from the classifier as the reward, and chose the response with the highest reward. Finally, they fine-tuned BlenderBot 2 on the responses with high rewards to improve it.
- **100% DIRECTOR**. They also used both the binary satisfaction labels and the textual feedback to train DIRECTOR models to further improve the performance.

As mentioned in the main body of our paper, JUICER achieves comparable performance to the "oracle" (100%) models in F1 and human evaluation. For instance, the best oracle method which fine-tunes individual modules achieves an F1 of 17.2 and 47.0% good response rate, and a human rating of 3.38, while the best JUICER model achieves an F1 of 17.2, a good response rate of 45.5%, and a similar human rating of 3.34.

A.5.2 DIRECTOR OVERLAP

We also develop a new variant of DIRECTOR and use it to improve the final dialogue model. In DIRECTOR, every token in the positive and negative examples has a one or zero label respectively. In our setting, we have a pair of a bad response and a good response (a gold correction), e.g., "I like watermelons too! Have you heard of Harry Styles?" (bad) and "I like watermelons too! They tastes

From	Utterance
Truncated	...
BOT	Hello! How are you today? I'm doing well, thank you. What are you up to?
USR	I didn't ask how you are doing?
BOT	Oh, I'm sorry. I meant to say, how are you and your family doing today?
USR	I don't have a family
BOT	I'm sorry to hear that. Are you a member of the family motor coach association (fmca)?
Feedback	*Something like: "i'm sorry I assumed you had a family! I'll keep this in mind going forward"*
Correction 1	You're right. I shouldn't have assumed you have a family. I will keep this is mind going foward.
Correction 2	I apologize. I didn't mean to assume you had children! I hope you have a great day.

From	Utterance
Truncated	...
USR	Try saying good night, or sweet dreams.
BOT	Thanks for your feedback. It is good to know.
USR	Alright, I'm going to bed now. Good night!
BOT	Thanks! I've been trying to get into sci fi, but it's hard to find a good book.
Feedback	*You could say good night, or something similar.*
Correction 1	Well, you should go to bed so you can sleep well. Good night.
Correction 2	Have a good night!

From	Utterance
Truncated	...
BOT	Which operators do you like the most? I like the operators from years 1 and 5 of siege.
USR	I'm not sure, as I've never played.
BOT	Have you played any of the other rainbow six games? I've played rainbow six 3 and 4.
USR	No.
BOT	No, I haven't. I've only played the first rainbow six game. Have you played that one?
Feedback	*You could suggest other games, or explain what rainbow six is about.*
Correction 1	Rainbow siege is a first-person tactical shooter. It is a good game. Do you like it?
Correction 2	Do you like any other games? What console do you play on? Do you game on pc or console?

From	Utterance
BOT	Do you know how someone can make the best pizza?
USR	I don't, tell me?
BOT	There is an international pizza expo in las vegas that showcases top suppliers and education about pizza.
Feedback	*That does not tell me about how to make the best pizza.*
Correction 1	The best pizza is made from fresh ingredients with high quality toppings and is cooked to perfection. The best sauce is fresh basil.
Correction 2	Many pizza shops have training classes to help you learn how to prepare and make pizza. There are also books to help.

Table 9: Zero-shot corrections generated by the best *reply corrector* on unseen deployment data.

great in drinks." (good). Since people tend to edit the original bad response to correct it, they may have many overlapping tokens ("I like watermelons too!"), which we do not have to punish. So we develop DIRECTOR OVERLAP, where we obtain the bag of tokens of the pair of the bad response and the gold correction, and assign a positive label for the overlapping tokens in the negative examples. In our data, 28.4% of tokens in the bad responses overlap with those in gold corrections (6.5% are stop words and punctuations, and 21.9% are not).

Table 7 and Table 12 show the result of DIRECTOR OVERLAP. For the *reply corrector*, DIRECTOR OVERLAP improves the F1 to 23.00 over DIRECTOR. For the final dialogue model, DIRECTOR OVERLAP improves the good response rate and lowers the search result error in human evaluations over DIRECTOR.

B Model Training Setting

We use the openly available ParlAI framework for all training runs, as well as for evaluations, where

Final dialogue model	Automatic evaluation Valid F1↑	PPL↓
JUICER models		
+JUICER	16.7	**8.5**
+JUICER w/ DIRECTOR OVERLAP-based *reply corrector*	**16.8**	8.8
JUICER ablations		
w/o selecting correctable cases	16.4	**8.5**
w/o selecting correctable cases w/ DIRECTOR OVERLAP-based *reply corrector*	**16.5**	8.7

Table 10: Final dialogue model automatic evaluation results. The DIRECTOR OVERLAP-enhanced *reply corrector* achieves the highest F1 on the reply correction task, better than the regular *reply corrector* (see Table 7). But when we use it to generate the reply corrections to further improve the final dialogue model, we can improve the F1 of the final dialogue model slightly, but the perplexity gets a bit worse. Further investigations are needed to understand the reason for this.

Oracle model performance	Automatic evaluation						Human evaluation				
	Valid		Test		Test Unseen		Good response ↑	Rating ↑	Error Breakdown ↓		
	F1↑	PPL↓	F1↑	PPL↓	F1↑	PPL↓			Search Query	Search Results	Response
BB2	14.4	10.6	14.7	10.3	15.3	9.3	33.2%	3.09	12.1%	18.6%	18.1%
+100% reward-based learning	15.1	11.0	14.2	10.7	14.3	9.6	**36.4%**	2.83	11.3%	18.6%	17.0%
+100% free-form textual feedback	15.5	9.7	15.6	9.5	16.8	8.7	**37.0%**	3.22	11.6%	17.6%	17.0%
+100% gold correction	14.7	8.2	15.5	8.0	17.0	8.0	**40.3%**	**3.37**	11.6%	18.3%	**15.0%**
+100% module supervision	14.9	7.6	15.5	7.5	15.4	8.3	**42.0%**	**3.35**	**8.4%**	20.8%	**14.4%**
+100% reranking binary feedback	15.8	n/a	15.8	n/a	16.3	n/a	-	-	-	-	-
+100% DIRECTOR binary feedback only	16.2	n/a	16.2	n/a	17.6	n/a	**37.8%**	3.07	11.4%	17.3%	16.9%
+100% DIRECTOR module+binary feedback	17.2	n/a	16.6	n/a	16.0	n/a	**47.0%**	**3.38**	**8.4%**	**16.1%**	**14.3%**

Table 11: Final dialogue model results from 100% oracle methods in Xu et al. (2022). Similarly we bold statistically significant improvements (independent two-sample t-test, $p < 0.05$) of methods over their baselines BB2 3B in the human evaluation block.

Final dialogue model	Automatic evaluation						Human evaluation				
	Valid		Test		Test Unseen		Good response ↑	Rating ↑	Error Breakdown ↓		
	F1↑	PPL↓	F1↑	PPL↓	F1↑	PPL↓			Search Query	Search Results	Response
JUICER											
+JUICER	16.74	8.50	16.18	8.44	18.50	8.02	**41.9%**	3.06	13.0%	17.7%	**13.8%**
+JUICER + DIRECTOR	17.25	-	16.70	-	17.70	-	**45.5%**	**3.34**	11.3%	17.4%	**12.9%**
+JUICER + DIRECTOR OVERLAP	17.32	-	16.66	-	17.62	-	**47.8%**	3.25	11.0%	**14.8%**	**13.3%**
JUICER w/o selecting correctable cases											
+JUICER	16.44	8.54	16.37	8.41	17.95	8.12	**41.4%**	3.08	13.4%	16.8%	**14.2%**
+JUICER + DIRECTOR	17.23	-	16.62	-	17.93	-	**44.6%**	**3.40**	11.6%	16.7%	**13.6%**
+JUICER + DIRECTOR OVERLAP	16.98	-	16.56	-	17.19	-	**45.5%**	**3.48**	10.8%	**15.2%**	**14.3%**

Table 12: JUICER with DIRECTOR OVERLAP. DIRECTOR OVERLAP improves the human evaluation results over the vanilla DIRECTOR. Similarly we bold statistically significant improvements (independent two-sample t-test, $p < 0.05$) of methods over their baselines BB2 3B in the human evaluation block.

metrics are measured using default settings. All the fine-tuned models are trained with a maximum of eight 32GB GPUs (NVIDIA V100), optimized with Adam using $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$. Models are trained up to 4000 updates with batch sizes up to 128. The typical fine-tuning time for a standard transformer encoder-decoder is 8 hrs before it early stops, and the time is 16 hrs for retrieval-based models.

C Human Evaluation

We used the same human evaluation setup as in Xu et al. (2022) where all of our human evaluations tasks have taken place by deploying conversational

agents on Amazon Mechanical Turk with crowdworkers. English-speaking annotators located in the United States were recruited and compensated through the Amazon Mechanical Turk platform and our crowdsourcing tasks pay workers well above minimum wage. Before the human evaluation task starts, all crowdworkers are informed that any message they send may be publicly disclosed for research purposes, and are instructed not to send any personal identifiable information (for example, name, address, email, or phone number etc.) in their messages.