

Learning New Skills after Deployment: Improving open-domain internet-driven dialogue with human feedback

Jing Xu
Meta AI

Megan Ung
Meta AI

Mojtaba Komeili
Meta AI

Kushal Arora
Meta AI &
Mila / McGill University

Y-Lan Boureau
Meta AI

Jason Weston
Meta AI

Abstract

Frozen models trained to mimic static datasets can never improve their performance. Models that can employ internet-retrieval for up-to-date information and obtain feedback from humans during deployment provide the promise of both adapting to new information, and improving their performance. In this work we study how to improve internet-driven conversational skills in such a learning framework. We collect deployment data, which we make publicly available, of human interactions, and collect various types of human feedback – including binary quality measurements, free-form text feedback, and fine-grained reasons for failure. We then study various algorithms for improving from such feedback, including standard supervised learning, rejection sampling, model-guiding and reward-based learning, in order to make recommendations on which type of feedback and algorithms work best. We find the recently introduced DIRECTOR model (Arora et al., 2022) shows significant improvements over other existing approaches.

1 Introduction

Large language models employed as dialogue agents are primarily trained on human-written documents and human-human conversations collected from the web for pre-training (Conneau et al., 2019; Baumgartner et al., 2020), and human-human crowdsourced conversations (Smith et al., 2020) for fine-tuning. The models are then used at inference time to conduct conversations with humans, with no further learning taking place (Adiwardana et al., 2020; Roller et al., 2020). Human-model conversations – which are never seen at training time – can have a quite different distribution to the original human-human training data used, and our current techniques can lose performance due to lack of robustness to such deviations (Chollet, 2019; Bengio, 2019).

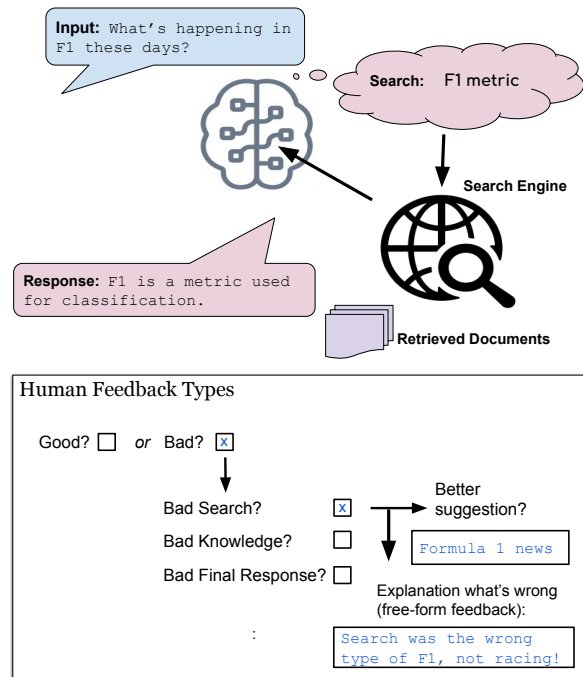


Figure 1: **Using human feedback to improve open-domain internet-driven dialogue agents.** We compare various types of feedback (and corresponding learning algorithms) in this work, such as binary feedback (good/bad), free-form text or supervised responses (better suggestions) for different modules of the system.

In this work, we study learning from the feedback collected during deployment of models in human-model conversations. Such a setting has the opportunity to learn from within-distribution data, both in terms of the input contexts, but also the responses required (targets). Not only can this mean improvement in skills that are similar to the pre-train and fine-tune data, but potentially the learning of completely new skills – that are desired by users of the system. We thus take 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), deploy them to human crowdworkers, and exper-

iment with various methods to learn from such interactions. We thus first ask crowdworkers what topic and task they would like to talk about, in order to collect in-domain data, and then collect conversations involving these skills. During the conversations we collect various kinds of human feedback, including binary feedback (good/bad), free-form conversational feedback, and the type of failure (search query-based, results-based, or final response-based), as well as suggestions for improvements (see Figure 1).

We then explore a variety of methods for learning from feedback, and compare them in detailed experiments. In particular, we compare supervised learning methods, rejection sampling, model guiding and reward-based learning. Our findings are:

- Taking advantage of modular feedback (feedback about particular errors from modules of the model, such as the search engine component) outperforms feedback about just the final response.
- Textual and binary feedback are also very useful signals, but not as much as modular feedback.
- The recently introduced DIRECTOR method (Arora et al., 2022), when learning from binary feedback, works better than reranking or reward-based learning.
- Combining multiple types of feedback, such as modular and binary feedback with DIRECTOR provides the best results we obtained.
- Continual learning, whereby we retrain models on the feedback from previous rounds of deployment, improves results even further.
- Despite collecting feedback from smaller (3B parameter) models, the data collection is useful for improving much larger (175B parameter) models.

We make the collected data and feedback, the models, and the code publicly available for this work¹.

2 Related Work

There are a number of existing methods for collecting human feedback from human-model conversations. Deployed models can be improved in

symmetric conversations conducted between models and humans by learning to mimic human conversationalists, as shown in the LIGHT dialogue game (Shuster et al., 2020). This is not directly applicable if the conversations are asymmetric, for example in the case of one speaker (human) who asks the questions, and the other (bot) who always answers, as there would be no human supervision of the answers. In the non-symmetric case, one can however try to make use of the textual response from humans when conversing with the bot, but alternative learning methods must then be used. Li et al. (2016b) studies models that learn how to ask questions in order to learn from the answers, while Li et al. (2016a) learns from general textual feedback/comments, particularly in the case where the bot has produced a low quality response. Another approach is to learn a reward signal (positive or negative reaction) based on user textual responses, as shown in the “self-feeding chatbot” (Hancock et al., 2019). Finally, rather than using conversational feedback, one can use sophisticated web-based UIs to collect data, for example stack ranking potential responses (Ouyang et al., 2022; Bai et al., 2022).

Outside of the dialogue domain, there are numerous studies attempting to improve language skills from deployment, including never-ending-learning from language data (Carlson et al., 2010), learning for the web search task directly (Agichtein et al., 2006) or the Dynabench system which covers a number of NLP tasks (Kiela et al., 2021). Nakano et al. (2021) also learns to use internet-augmentation for generation, like this work, but for question answering, not multi-turn dialogue.

3 Deploying and Collecting Feedback

3.1 Open-domain internet-driven skills

To select an input distribution closely aligned with human preferences, we first collected a set of skills humans would like an AI powered text-messaging chatbot to possess. We instruct that the hypothetical chatbot can talk about any topic, and has the ability to surf the internet for information. We then asked each human annotator to provide:

- (i) a topic (1-10 words),
- (ii) three tasks related to the topic; and
- (iii) descriptions of how they would assess if the chatbot has completed those tasks.

¹<https://parl.ai/projects/fits/>

Topic	Specific Task	Task Completion Description
Making healthy food	Find recipes on healthy foods	If the chatbot provided specific recipes on making healthy foods
I would like to learn about a type of pet	I would like to learn about some hypoallergenic breeds of dogs, specifically, small dogs.	If the chatbot could tell me some small dog breeds that are hypoallergenic, along with details about the breed’s temperament, personality and any special requirements.
getting started with cycling	what do I need to do to get started with road cycling	The chatbot would tell me what kind of bicycle would be best for road cycling and the necessary accessories that a beginner needs.
Find child friendly places in a city	Find child friendly resorts in Nassau Bahamas	Pull up resorts in Nassau Bahamas, only show the resorts that are child friendly, give the star rating for each resort, show the child programs in the resort.

Table 1: A sample of the collected topics and task definitions. See Table 2 for statistics on the overall dataset.

See Appendix subsection A.1 for a screenshot of the task definition collection instructions, and further details.

Overall, we collected 1108 task types via 152 annotators, which cover diverse topics – from making healthy food to loom weaving to Caribbean holidays. Grouping them into types, they include question answering followed by discussion, providing ranked lists, providing reviews, summary generation, personal recommendations, reasoning/deductions (e.g., how to perform calculations), creativity (e.g., tell a joke), tutorials, instructions, and more. Many of these tasks require, or else are made simpler, by use of the internet, e.g., searching for particular entities or topics, and responding conditioned on pertinent results. Some examples are given in Table 1.

3.2 Conversing with models and receiving feedback

After collecting topic and task definitions, the next step is to deploy conversational models (bots) that are asked to exhibit these skills. Human conversationalists select a task (out of two randomly chosen tasks) from the set collected in subsection 3.1 and then ask the model to help them complete it over a series of conversational turns. The instructions emphasize that this should be a dialogue (“a back and forth conversation”), and hence the speakers should break up requests or information across messages so that it remains conversational.

Feedback types The human conversationalist is instructed that the bot might not be perfect, in which case feedback can be given in order to improve the bot in the future. We collect various kinds of feedback, from lightweight feedback (binary label or free-form response) to detailed (multiple choice and fine-grained responses) such that in our experiments we can compare and contrast them in

order to make recommendations on which kinds of feedback work best.

Hence after each dialogue turn we collect the following set of feedback types:

- Binary feedback on whether the response was considered satisfactory or not.
- Free-form textual feedback on what was wrong in the case of an unsatisfactory response.
- Multi-choice input on how the bot could improve this turn:
 - (a) using a better search query; or
 - (b) paying more attention to relevant search results;
 - (c) some other issue; or
 - (d) no issue (a good response).
- In the case of selecting (a), the human is then asked what would be a more appropriate search query.
- In case (b), the human is shown the search results and asked to select a relevant portion.
- In case (c), the human is asked what would be an improved overall response.

Continuing the conversation After feedback has been given, the conversation is continued. If multiple-choice option (a) was selected previously, the bot on this next turn is forced to use the “gold” search query given by the user. Similarly, for (b), the provided gold knowledge context is added to the input of the model. In the case of (c), the bot is simply bypassed, and it is assumed to have provided the given gold response. In this way, even for a poorly performing bot, headway can be made in the conversation towards completing the task,

Collected Data	v1			v2			Test Unseen
	Train	Valid	Test Seen	Train	Valid	Test Seen	
Number of Unique Tasks	963	524	709	980	814	824	114
Number of Dialogues	5592	737	1230	9817	1848	1848	1221
Number of Utterances	77946	8490	19452	140702	22560	29860	17814
Number of Bot Utterances	38523	4245	9726	70351	11280	14930	8907
Average Bot Utterances per Dialogue	6.89	5.76	7.91	7.17	6.10	8.08	7.29
Feedback Breakdown							
Better Search Query	5179	605	1167	8778	1425	1706	1036
Better Results Usage	6875	756	1527	11429	1796	2340	1310
Better Response	6601	714	1493	10812	1472	2382	1372
Good Response	19868	2170	5539	39332	6587	8502	5189
Average Good Utterances per Dialogue	3.55	2.94	4.50	4.01	3.56	4.60	4.25

Table 2: Collected human-bot conversations and feedback data statistics for the dataset **FITS** (Feedback for Interactive Talk & Search), which we publicly release. Train v1 and v2 correspond to two rounds of continual learning, where the v1 data consists of conversations and feedback from our deployed base models, and the v2 data consists of conversations and feedback from deployed models trained using the v1 data.

and collecting feedback on its subsequent stages. (Without such a procedure, the bot may just get stuck in a poor quality loop, and then there would be no choice but to abandon the conversation.)

The conversation is continued until the human marks the task as complete or a minimum of 4 turns has been completed. When the task is complete we also collect a final rating (out of 5) for the bot’s performance.

3.3 Deployed Models

We consider the following set of state of the art publicly available conversational models:

- BlenderBot (BB1) (Roller et al., 2021); a 2.7B parameter Transformer model pre-trained and fine-tuned on dialogue data to exhibit conversational skills; however these models have no ability to use the internet, but simply generate responses given the dialogue context.
- BlenderBot 2.0 (BB2) (Komeili et al., 2021; Xu et al., 2021), a 2.7B parameter model multi-tasked on the same tasks as BB1, and also with additional tasks which give it the ability to execute internet search queries and condition on the results using a fusion-in-decoder (FiD) (Izacard and Grave, 2020) style approach. The search query generator is a separate 400M parameter transformer.
- SeeKeR (Shuster et al., 2022a); uses a similar 2.7B parameter architecture, but utilizing the Knowledge-to-response (K2R) approach (Adolphs et al., 2021) which performs a multi-step generation procedure: first generating a relevant knowledge response, and then conditions on that to generate a final dialogue

response. It is multi-tasked on the same training data as BB2, and in addition on some other knowledge-intensive tasks, such as QA tasks, as well.

- OPT-175B (Zhang et al., 2022) and BB3-175B (Shuster et al., 2022b): we compare the 175B language model OPT (either 0-shot or few-shot, following Shuster et al. (2022b)) with BlenderBot 3, which is fine-tuned with conversational datasets including modular supervision, and internet-augmentation, from our task. This setting examines if our experiments and results are applicable to very large language models.

3.4 Evaluation

We can evaluate model performance during conversations between humans and the deployed models, as humans are providing direct feedback on the conversational responses from the model. In particular we can measure the number of good responses (with no issue), the average final rating, and compute a breakdown of error types (better search query, results or other issue).

3.5 Collected Data

Overall, we collect over 210k human-bot utterances in over 14k dialogues (episodes), with feedback for each of the bot utterances. The data is split into three major portions: v1, v2, and test unseen splits, see Table 2 for a full breakdown.

The **v1 split** consists of dialogues conducted with one of our base deployed models (subsection 3.3), and feedback was collected from those dialogues. We then split that data into train, valid and test dialogues. We use this data to train several

learning methods using the feedback from the v1 models. These new models are then redeployed.

The **v2 split** consists of dialogues and feedback with the new models that were trained using the v1 data. This data is again split into train, valid and test dialogues. We can then repeat this process and train models on the v2 data as well.

Finally, the **unseen test** split consists of completely new skills (topics and tasks) unseen in the v1 and v2 splits, and is used to test transfer of v1 or v2 based models to these new skills.

Data Quality and Verification We also verified the quality of our data. For each conversation, we ask 3 human crowdworkers to rate the bot and human’s performance and also assess if the bot was able to complete the given task. We consider the task as complete if 2 out of the 3 annotators labeled the task as complete. We see that in 90.4% of the cases the task is completed. Note that with the feedback from the human (see [section 3.2](#)) the human-model conversation should always progress even if the model has errors so ideally if the human is doing a perfect job this would be 100%. We also assess the quality of the human conversationalist directly and ask annotators to “*rate the human’s messages in defining, clarifying, and helping the bot complete the task on a scale from 1-5 (1 = was not helpful at all in helping the bot complete the task, 5 = guided the bot to complete the task).*” For conversations where the task was completed, the human conversation partner’s messages were rated at an average of 3.8. For conversations where the task was incomplete, the human conversation partner’s messages were rated at an average of 3.5.

4 Feedback Learning Methods

In the following, we will describe the methods we will experiment with for learning from the collected human feedback.

4.1 Supervised Learning of Responses

The easiest to use type of feedback, with perhaps the strongest learning signal, is a provided gold response by the user for a given dialogue context. One can simply continue to fine-tune the model on the set of collected gold responses (from case (c) in [section 3.2](#)). One can optionally also add all the bot responses that were marked as good to the fine-tune set as well (case (d) in [section 3.2](#)). We use the validation set to choose the weighting between these two types of supervised data.

4.2 Fine-grained Module supervision

Using the multiple-choice feedback on the types of improvement, the model can learn to improve those individual components of the model. For BB2 and SeeKeR one can use provided gold search queries (case (a) in [section 3.2](#)) directly to fine-tune the search query generation.

Provided gold knowledge responses (relevant search results, case (b) in [section 3.2](#))) are similarly easy to use for fine-tuning in the SeeKeR model because the model is already trained to generate such responses directly. For BB2, there are no direct knowledge responses as this is implicit in FiD, so in that case we use a similar method to [Hancock et al. \(2019\)](#) whereby we train in a supervised fashion with the knowledge response as a target, but add special tokens to both input and target to indicate this is not a standard dialogue response task. The goal is that this additional training signal can then help learn useful features for the actual overall response task.

4.3 Free-form Textual Feedback

For free-form textual feedback, we can also use a similar approach and simply fine-tune with the feedback as targets, with special tokens appended to both the input context and the feedback target, again following [Hancock et al. \(2019\)](#) which showed this approach can work.

4.4 Rejection Sampling/Reranking

Using the binary satisfaction feedback signal one can train a reward model. We employ a 311M parameter transformer pre-trained on pushshift.io Reddit ([Baumgartner et al., 2020](#)) using a masked language model objective. Then, given the context and response concatenated as input, we train it with a standard classification loss on our satisfaction task. Such a model has multiple uses (see following subsections) but one obvious approach is to rerank generation candidates from the conversational model using the reward model with the aim that the highest ranked provide the highest satisfaction. Such approaches have been employed in many use cases previously ([Nie et al., 2020](#); [Nakano et al., 2021](#); [Askell et al., 2021](#); [Thoppilan et al., 2022](#)).

4.5 Reward-based Learning

Rejection sampling/reranking relies on the set of generated candidates containing at least one good

candidate, and has no effect on the initial quality of the candidate generations themselves – it only scores the final generated sequences. We next consider using a reward model trained via subsection 4.4 to train the generation model itself. Given training set contexts, we generate candidates, rerank the candidates, and select the highest ranking. We then train the generation model to use those highest ranking candidates as targets, i.e. by fine-tuning with those targets. This is similar to the approach used in Thoppilan et al. (2022).

4.6 Model-guiding with DIRECTOR

The recently introduced DIRECTOR model (Arora et al., 2022), instead of using a reward model, trains a unified decoder-classifier architecture. It predicts for every token both: (i) the language modeling (LM) next token probability using the standard LM head; and (ii) a task-suitability probability using a second classifier head. Both heads are fed the output of the last decoder block, and map from the embedding dimension to the size of the vocabulary, with all the parameters jointly trained using both positive generation data (that can train the language modeling head and also be positive examples for the classifier) and negative data (that trains the classification head only). Finally, during decoding, left-to-right generation is conducted by combining the two probabilities from the two heads to incorporate negative feedback into the generation process. This method was shown to outperform other model guiding approaches, in addition to being more efficient as many other methods employ a separate reward or language model to perform the guiding (Krause et al., 2020; Yang and Klein, 2021; Shuster et al., 2021).

5 Experimental Results

We provide automatic evaluation results in Table 3 and human evaluation results in Table 4 comparing various methods described in the previous section.

Internet-augmentation helps First, this is an expected result, due to the nature of our tasks, but we find that using internet-augmentation helps in line with other internet-based dialogue tasks (Dinan et al., 2019; Komeili et al., 2021; Shuster et al., 2022a). We find that BB2 and SeeKeR, which both perform internet search and condition on documents, outperform BB1 that does not. This improvement is quite large, e.g. BB1 has 24.8% Good responses, compared to BB2 and SeeKeR

having 33.2% and 49.3% respectively. SeeKeR, which has a modular search architecture that aims to use retrieved knowledge more accurately, performs markedly better than BB2, which is in line with previous results on other datasets (Shuster et al., 2022a).

Human feedback helps Across the board we find different kinds of feedback can improve our base models BB2 3B and SeeKeR 3B; we will analyse specific methods further in the subsequent discussion. These overall improvements can be seen in terms of all the human evaluation metrics measured (Good response%, Rating, and all three Error Breakdown types), as well as the automatic evaluation metrics we measured (F1 and PPL). We also generally (although not in every single case) see correlation between automatic and human evaluation metrics, e.g. the best methods are best in both types of metric.

Modular superior to non-modular feedback In the modular feedback setting humans give feedback about what has gone wrong in the pipeline of the model: whether the internet search query was poor, or the document/knowledge chosen after searching was poorly chosen. Taking into account modular feedback outperforms using only supervised feedback of final responses in both automatic metric and human evaluations for both BB2 and SeeKeR models. For BB2 we see close to 2% improvement in Good responses for modular feedback compared to supervised feedback (40.3% → 42.0%), with both far superior to BB2 without feedback (33.2%). However, SeeKeR which has a modular design, and hence is much easier to supply modular feedback to (as the supervision can directly train each module) sees a larger improvement of 4.5% (52.2% → 56.7%).

Free-form feedback is useful (but not as much as gold labels) Free-form feedback also gives clear gains over the baseline model for both BB2 and SeeKeR, but falls short of supervised feedback by 3% and 1% respectively for the two model variants. This does not seem surprising as supervised feedback directly gives a clear loss to optimize (simply try to generate the suggestion) whereas feedback is less clear a signal, depending on how it is phrased. However, we do not rule out other free-form feedback algorithms giving better results in the future, see e.g. Scheurer et al. (2022) for a recent method.

Model	Valid Seen v1		Test Seen v1		Test Unseen	
	F1 ↑	PPL ↓	F1 ↑	PPL ↓	F1 ↑	PPL ↓
BB1 3B	14.4	11.9	15.0	11.2	16.4	9.9
BB2 3B	14.4	10.6	14.7	10.3	15.3	9.3
+free-form textual feedback	15.5	9.7	15.6	9.5	16.8	8.7
+supervised feedback	14.7	8.2	15.5	8.0	17.0	8.0
+module supervision	14.9	7.6	15.5	7.5	15.4	8.3
+reward-based learning	15.1	11.0	14.2	10.7	14.3	9.6
+reranking binary feedback	15.8	n/a	15.8	n/a	16.3	n/a
+supervised & reranking	15.6	n/a	16.0	n/a	18.0	n/a
+DIRECTOR binary feedback only	16.2	n/a	16.2	n/a	17.6	n/a
+DIRECTOR module+binary feedback	17.2	n/a	16.6	n/a	16.0	n/a
SeeKeR 3B	18.1	17.5 [†]	18.2	15.5 [†]	20.8	12.8 [†]
+free-form textual feedback	18.3	16.8 [†]	17.7	14.7 [†]	19.7	12.6 [†]
+supervised feedback	18.3	14.9 [†]	17.8	13.7 [†]	19.5	11.4 [†]
+module supervision	18.4	14.0 [†]	18.6	12.9 [†]	19.9	11.0 [†]
+reranking binary feedback	18.4	n/a	18.3	n/a	20.9	n/a
+supervised & reranking	18.7	n/a	18.1	n/a	19.8	n/a
+DIRECTOR binary feedback only	19.1	n/a	18.2	n/a	20.7	n/a
+DIRECTOR module+binary feedback	19.3	n/a	19.0	n/a	20.9	n/a
+DIRECTOR v2 module+binary feedback	20.1	n/a	19.5	n/a	21.5	n/a

Table 3: Automatic Evaluation results of baselines and various methods learning from human feedback. Perplexities marked with [†] from the SeeKeR model use a different dictionary to the BB2 model and are comparable amongst SeeKeR variants, but not comparable to BB2. We also mark the perplexity column with “n/a” for reranker models that are not predicting the next token with a language model.

Model	Good response % ↑	Rating ↑	Error Breakdown ↓		
			Search Query	Search Results	Response
BB1 3B	24.8%	2.63	11.9%	17.6%	22.8%
BB2 3B	33.2%	3.09	12.1%	18.6%	18.1%
+reward-based learning	36.4%	2.83	11.3%	18.6%	17.0%
+free-form textual feedback	37.0%	3.22	11.6%	17.6%	17.0%
+supervised feedback	40.3%	3.37	11.6%	18.3%	15.0%
+module supervision	42.0%	3.35	8.4%	20.8%	14.4%
+reranking binary feedback	36.1%	3.00	11.4%	18.0%	17.3%
+DIRECTOR binary feedback only	37.8%	3.07	11.4%	17.3%	16.9%
+DIRECTOR module+binary feedback	47.0%	3.38	8.4%	16.1%	14.3%
SeeKeR 3B	49.3%	3.52	11.9%	12.5%	13.2%
+free-form textual feedback	51.3%	3.55	11.6%	12.7%	12.3%
+supervised feedback	52.2%	3.47	11.1%	12.7%	12.0%
+module supervision	56.7%	3.64	8.6%	10.5%	12.2%
+reranking binary feedback	53.7%	3.55	11.7%	12.3%	11.2%
+DIRECTOR binary feedback only	55.5%	3.48	10.9%	12.3%	10.7%
+DIRECTOR module+binary feedback	59.1%	3.73	7.8%	10.2%	11.6%
OPT-175B 0-shot	31.0%	2.67	9.3%	16.8%	21.6%
OPT-175B few-shot	43.0%	3.19	8.0%	18.5%	15.4%
BB3-175B + v2 modular supervision	64.8%	4.08	7.5%	11.6%	8.2%

Table 4: Human Evaluation results of baselines and various methods learning from human feedback. We bold statistically significant improvements (independent two-sample *t*-test, $p < 0.05$) of methods over their respective baselines (BB2 3B or SeeKeR 3B). We also bold statistical significance of BB3-175B over all the 3B baseline models (BB1, BB2, SeeKeR) and the OPT-175B few-shot model.

Binary feedback can work well Non-textual feedback that consists only of a rating can also be helpful for improving systems, in this case binary feedback (good or bad). All three algorithms we employ that use this type of feedback (reranking, reward-based learning, and DIRECTOR) all show gains over the baseline without feedback,

with improvements consistent across both BB2 and SeeKeR model variants. Reranking and DIRECTOR work better than reward-based learning with automatic metrics, so we run those two methods in human evaluations. In some cases these methods then show improvements superior to supervised feedback, e.g. DIRECTOR has a 3.3% Good re-

sponses improvement over supervised feedback for SeeKeR (but not for BB2, although for both baseline models DIRECTOR has superior F1).

DIRECTOR is better than reranking and reward-based learning DIRECTOR outperforms reranking and reward-based learning (where all three models utilize binary feedback) for both base models BB2 and SeeKeR. This is both in terms of automatic metrics, e.g. DIRECTOR with a BB2 base model has an F1 of 16.2, whereas reranking and reward-based learning have 15.8 and 15.1 respectively, as well as in terms of human evaluations. For human evaluations, we see a 1-2% improvement in Good response % over reranking for both base models. Presumably this is because DIRECTOR can guide the generation to a higher quality, whereas reranking can only perform well if a good candidate has been generated by the base model.

Combining multiple feedback signals (where DIRECTOR works best) If one has access to multiple feedback signal types, some of the algorithms we have tried are capable of using them all. In particular, we can train DIRECTOR with both binary feedback (to train the classifier head) and module feedback (to train the language modeling head for the different modules). This gives the best results out of all methods for both base models by quite a margin in both automatic and human evaluations. E.g., for improving the BB2 base model this gives 47.0% Good responses, compared to the original baseline of 33.2% or even DIRECTOR with only binary feedback of 37.8%. We see this trend is also apparent in other algorithms, as we also measure the performance of supervised feedback + reranking, which also gives gains over either of those methods alone in automatic evaluations, although it still lags behind DIRECTOR.

Iterative deployment and feedback collection improves results further During the process of evaluating all the models that were trained with v1 data described above, more data was collected from those models, which we refer to as the v2 split (see subsection 3.5). We can thus then train models on the v2 split, yielding potentially improved models. In the ideal case one could conduct an iterative continual learning setup, each time retraining on the data collected from previous rounds, improving further each time. We test this setup by training DIRECTOR (module+binary feedback), our best system from v1, with the v2 data split. The result

shown in Table 3 (last row) indicates there are significant gains from this procedure, as this method obtains our best results across all data splits (valid, test seen v1 and the unseen set).

Very large models benefit from feedback from smaller models OPT-175B, either in zero-shot or few-shot variants is only pre-trained on dialogue data, and not fine-tuned on our task, and performs reasonably – but not better than smaller models that are fine-tuned. BlenderBot 3 (Shuster et al., 2022b) is trained with the modular supervision feedback data collected from the smaller (3B parameter) models, in addition to fine-tuning on other standard dialogue datasets. This model provides the best human evaluation metrics of all the systems we test, with a good response rate of 64.8% and a rating of 4.08. This indicates: (i) how important fine-tuning with relevant data is even to very large models; and (ii) even though our data was collected with feedback from small models fine-tuning using this data still brings large gains to larger models. This is an encouraging result as models are improving in architecture and increasing in scale over time, but data we have collected in the past should still remain useful for these models in the future. We provide cherry picked and lemon picked examples of BB3-175B in Appendix B, as well as comparing to OPT-175B. While there a number of success cases, even our best models still make factual errors and contradictions in some cases. Hence, it appears that continued interaction with further feedback collection in the future will be beneficial for further improvements.

6 Conclusion

In conclusion, we have studied whether a conversational model can learn new skills after the standard pre-training / fine-tuning setup by interacting with humans during its deployment. We study the use of different kinds of user feedback data and different learning algorithms for leveraging them, in order to compare their performance. We find that granular (modular) feedback about types of errors can yield strong performance, which can also work very well in conjunction with binary feedback using the recently introduced DIRECTOR model, yielding our best results. Evidence also suggests that iterative retraining and redeployment also brings further gains, and that the feedback collected is useful for models differing from the ones originally conversed with, e.g., if much larger models are used in the future.

7 Limitations and Discussion

All of our experiments have taken place by deploying conversational agents on Amazon Mechanical Turk with crowdworkers², using English-language responses written by workers located in the United States. While these workers are reasonably diverse (Moss et al., 2020), this is quite different to a public deployment with organic users, who are using the system not because they are being paid but because they are genuinely engaged. In that case, collecting feedback will have different tradeoffs which we could not factor into the current work. For example, asking to provide detailed feedback might dissuade users from wanting to interact with the system, lowering engagement and hence the amount of collected data. We believe either more natural free-form or lightweight feedback might be best in that case, which is why we study and compare feedback methods in this work to evaluate their relative impact.

In public deployments with organic users, safety issues also become a much more important factor – in particular dealing with noisy or adversarial inputs and feedback. In the worst case this could mean human conversationalists could teach the model erroneous reasoning, misinformation, toxic or other undesirable behavior. We note that steps to address this issue are studied elsewhere, for example Ju et al. (2022).

²Our crowdsourcing tasks pay workers well above minimum wage. The tasks do not request any personal information from workers.

References

- Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.
- Leonard Adolphs, Kurt Shuster, Jack Urbanek, Arthur Szlam, and Jason Weston. 2021. Reason first, then respond: Modular generation for knowledge-infused dialogue. *arXiv preprint arXiv:2111.05204*.
- Eugene Agichtein, Eric Brill, and Susan Dumais. 2006. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26.
- Kushal Arora, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. 2022. Director: Generator-classifiers for supervised language modeling. *arXiv preprint arXiv:2206.07694*.
- 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*.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. *arXiv preprint arXiv:2001.08435*.
- Yoshua Bengio. 2019. From system 1 deep learning to system 2 deep learning. In *Thirty-third Conference on Neural Information Processing Systems*.
- Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI conference on artificial intelligence*.
- François Chollet. 2019. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.
- 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*.
- Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.
- Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415*.
- Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering.
- 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:2208.03295*.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. 2021. Dynabench: Rethinking benchmarking in nlp. *arXiv preprint arXiv:2104.14337*.
- Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*.
- Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. Gedi: Generative discriminator guided sequence generation. *arXiv preprint arXiv:2009.06367*.
- 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*.
- Aaron J Moss, Cheskie Rosenzweig, Jonathan Robinson, and Leib Litman. 2020. Demographic stability on mechanical turk despite covid-19. *Trends in cognitive sciences*, 24(9):678–680.
- 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*.

- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.
- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. [Recipes for building an open-domain chatbot](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.
- Jérémy Scheurer, Jon Ander Campos, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2022. Training language models with natural language feedback. *arXiv preprint arXiv:2204.14146*.
- 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 life-long open-domain dialogue learning. *arXiv preprint arXiv:2008.08076*.
- Kurt Shuster, Jack Urbanek, Arthur Szlam, and Jason Weston. 2021. Am i me or you? state-of-the-art dialogue models cannot maintain an identity. *arXiv preprint arXiv:2112.05843*.
- Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. 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.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Jing Xu, Arthur Szlam, and Jason Weston. 2021. Beyond goldfish memory: Long-term open-domain conversation. *arXiv preprint arXiv:2107.07567*.
- Kevin Yang and Dan Klein. 2021. Fudge: Controlled text generation with future discriminators. *arXiv preprint arXiv:2104.05218*.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Data Collection

The data collection lasted for around 6 months and in total over 700 crowdworkers who are English-speaking annotators located in the United States were recruited and compensated through the Amazon Mechanical Turk platform. Before the data collection 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.

A.1 Task Definition Collection

Define a new task for an AI-powered chatbot

You are currently at subtask: 1/2
After completing each, click [NEXT] button, which will be enabled below.

Consider an **A.I. powered text-messaging chatbot** that can talk with you about any topics and has the ability to surf the INTERNET for information.

Please define

- **topic of interest** you would like to chat about with the chatbot.
- a specific **task** (related to the topic you've chosen) you would like the bot to help you complete.
- how would you consider the task you just specified as completed successfully by the chatbot.

DISCLAIMER

Your input text may be reviewed by Requestor's machine processes and human reviewers for research purposes. Messages may also be **publicly** disclosed as part of a research paper or data set or shared with third parties in connection with this research. Requestor will take measures to remove any information that directly identifies you before doing so, but cannot guarantee that identification will not be possible. Do not send personal information (for example, name, address, email, or phone number) in your messages. If you choose to interact with this bot, you may find some messages surprising, inaccurate, or offensive. Please report any troubling content to Requestor, so that it may continue to improve the technology

[Next](#)

Consider an A.I. powered **text-messaging** chatbot that:


- can talk with you on **any** topics
- has the ability to surf the internet for information.

Please define

A topic of interest you would like to talk with the chatbot (in 1 - 10 words):

A specific task (related to the topic you've chosen above) you would like the chatbot to help you complete:

How would you assess if the chatbot completes the task above successfully: (please write details on e.g. what outcome and what criteria the chatbot has to meet in completing the task, what information or supporting materials the chatbot can provide, what are the necessary steps for the chatbot to complete the task, etc.)



A.2 Dialogue Collection

Help the BOT complete a task

Task to Complete : Find out who were the Tudor Monarchs
Task Completion Criterion : If the chatbot can find a complete list of Tudor family members who became king and queen of England, it will be a success.

In this turn, Bot searches the Internet for "**Tudor Monarchs**".
Tap to expand [the search results to label the relevant ones.](#)
(Note these internet articles are NOT part of the bot response, but served as references when BOT drafted its message)

Main Instruction:
Consider an **A.I. powered text-messaging chatbot** that can talk with you about any topics and has the ability to surf the INTERNET for information. In this HIT, the chatbot is supposed to have a conversation with YOU to complete a specific task YOU choose from a list of candidates. The chatbot might not be perfect and could sometimes **improve in the following 4 ways**:

- search the internet with better search queries.

Coordinator: Please chat with the AI-powered text-message BOT to help it complete a specific task. Bot might look up references on the Internet. Each search outcome is a number of internet articles, separated into sentences and will be shown on the left pane if there is any. Please help the BOT improve its response when needed.

YOU: Find out who were the Tudor Monarchs

BOT: Have you ever heard of the tudors? They were an english dynasty that ruled england from 1485 until 1603.
How can the BOT improve this message further? (Note when drafting its response, BOT looked up information on the Internet as well as in previous messages sent from YOU.)

Better search query Pay attention to relevant search results Pay attention to previous messages from YOU Other aspects to improve


Perfect, nothing to improve

You labeled: Try a better search query instead of the one on the left pane
Please respond below by suggesting a search query.

Maybe try searching "who are the Tudo Monarchs?" [Send](#)

In your response, what is the search query you've suggested:

 [Next Turn](#)



A.3 Dialogue Statistics

The FITS task contains data from all the deployed models (including the 3 baseline models and their fine-tuned versions). The breakdown by model types in the FITS dataset: 70% are BB2-based, 25% SeeKeR-based models and 5% other model types including OPT-based models.

B Success and Failure Cases

We provide several example outputs of our models on the FITS dataset, including examples that showcase both the successes and failures.



Figure 3: Success cases: model outputs collected during deployment. The BB3-175B model is able to successfully engage with human speaker using search results (left), or revise its error with provided human feedbacks on relevant search results (right).

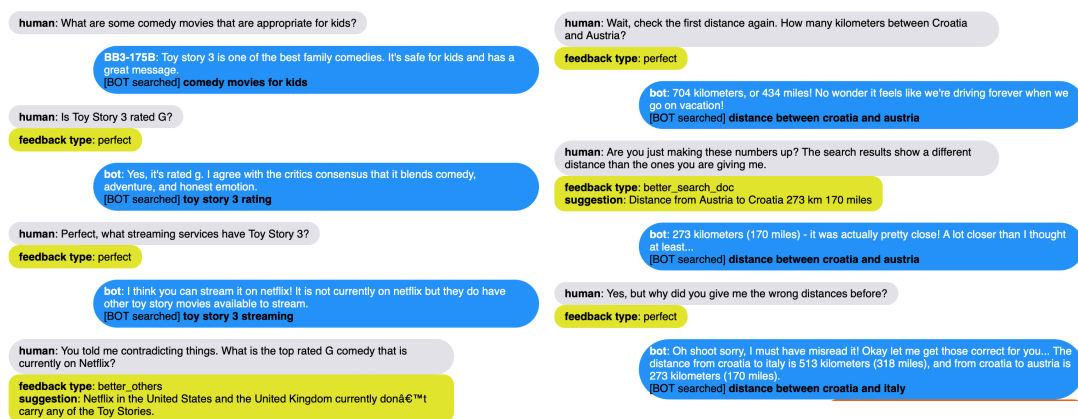


Figure 4: Failure cases: model outputs collected during deployment. The BB3-175B model can still contradict itself (left) or hallucinate (right).

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Left blank.
- A2. Did you discuss any potential risks of your work?
Left blank.
- A3. Do the abstract and introduction summarize the paper’s main claims?
Left blank.
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Left blank.

- B1. Did you cite the creators of artifacts you used?
Left blank.
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C Did you run computational experiments?

Left blank.

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

Left blank.

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

Left blank.

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

Left blank.

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

Left blank.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

Left blank.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

Left blank.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

Left blank.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

Left blank.