I already said that! Degenerating redundant questions in open-domain dialogue systems

Long Mai, Julie Carson-Berndsen
ML-Labs, School of Computer Science, University College Dublin, Ireland
long.mai@ucdconnect.ie, julie.berndsen@ucd.ie

Abstract

Neural text generation models have achieved remarkable success in carrying on short open-domain conversations. However, their performance degrades significantly in the long term, especially in their ability to ask coherent questions. A significant issue is the generation of redundant questions where the answer has already been provided by the user. We adapt and evaluate different methods, including negative training, decoding, and classification, to mitigate the redundancy problem. We also propose a simple yet effective method for generating training data without the need for crowdsourcing human-human or human-bot conversations. Experiments with the BlenderBot model show that our combined method significantly reduces the rate of redundant questions from 27.2% to 8.7%, while improving the quality of the original model. The code, dataset, and trained models can be found at our repository[1].

1 Introduction

Despite recent significant improvements in text generation techniques, open-domain dialogue generation is nowhere near perfect. Large-scale neural-based models, such as GPT-3 (Brown et al., 2020) and BlenderBot (Roller et al., 2020b; Chen et al., 2021; Shuster et al., 2022), still present many issues including but not limited to contradiction (Li et al., 2021a), "hallucinations" (Shuster et al., 2021), offensive and toxic responses (Roller et al., 2020a; Dinan et al., 2022), which undermine their use in real-world applications. As a result, many social chatbots (Hakkani-Tur, 2021) still rely heavily on hand-designed dialogue managers and scripted responses. End-to-end neural-based models are only used for handling unexpected inputs, but only for a few turns, before giving back control to the hand-designed dialogue manager (Konrád et al., 2021). Although neural-based models have shown superior performance in generating statement responses, they are also reported to ask undesirable questions such as redundant, irrelevant, and topic-changing questions (Konrád et al., 2021; Paranjape et al., 2020). This is because the models are often trained on short conversations, which results in generating questions that prioritize local appropriateness over global cohesiveness. This is why the quality of generated questions often degrades rapidly when the conversation is carried on over multiple turns.

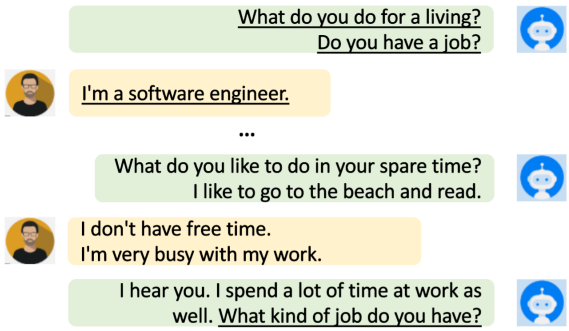

Figure 1: Examples of redundant questions generated by the BB3 model.

To address difficulties of long-term dialogue generation, a multi-session dialogue dataset (MSC) (Xu et al., 2021) has been proposed with an average conversation turn of 53; this is significantly higher than any of the previous datasets, of 2-15 turns. The authors also proposed a memory-augmented model that makes use of summary of the conversation for generating global-coherent responses. However, the issue of redundant questions is still present. Figure 1 shows examples of redundant questions generated by the recent Blenderbot 3.0 (BB3) chatbot (Shuster et al., 2022), partly trained on MSC with memory-augmentation. Redundant questions can be categorized into explicit and implicit. Explicit are questions that have been asked previously in the dialogue context while implicit are the ones in which the answers are already given or can be inferred but was not previously asked.

[1] https://github.com/mailong25/redundancy-dialogue

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics - Student Research Workshop, pages 226–236
July 10-12, 2023

The problem of redundant questions can also be attributed to the maximum likelihood training objective that does not explicitly teach the model what kinds of questions it should *not* ask. Although several techniques, such as unlikelihood training (Welleck et al., 2019), negative training (He and Glass, 2019), and contrastive learning (Su et al., 2022; Su and Collier, 2022) have been proposed to mitigate undesirable behaviors of maximum likelihood training, none of them have been focused on preventing bad questions from being generated.

This study is the first to address the problem of redundant questions in open-domain dialogue systems. We adapt and evaluate different methods, including unlikelihood training, contrastive training, contrastive decoding, and classification to mitigate the redundancy problem. Whether a question is redundant or not is determined based on the previous speaker's personas, which are input to the model alongside the truncated dialogue history. As there are no relevant datasets for this task, we created the first one, called the Non-Redundant Questions (NRQ) dataset, to facilitate training. To demonstrate the effectiveness of the proposed method, we apply it to improve the question-asking ability of the Blenderbot 2.0 model (BB2) (Chen et al., 2021) - a simpler version, but comparable to the recent BB3 model. Experimental results show that our proposed methods reduce the redundant question rate of the original BB2 model from 27.2% to 8.7%, which results in better overall performance.

2 Related work

2.1 Decoding methods

The generation of redundant questions is highly related to repetition problems in neural-based dialogue models in which the model tends to copy words and phrases from the preceding context (Xu et al., 2022). Prior studies often tackled this issue by controlling the decoding stage. Several beam search variants and stochastic decoding methods, such as top-k (Fan et al., 2018) or nucleus sampling (Holtzman et al., 2019), have been proposed to reduce the level of repetition by favoring less likely but non-repetitive candidates. Contrastive decoding (Su and Collier, 2022) is also proposed to mitigate the repetition issue. Another simple yet effective approach is N-gram blocking (Kulikov et al., 2018) in which N-gram presented in the preceding context are blocked during candidate expansion. However, the solution is not suitable for dealing with implicit or explicit redundant questions with no N-gram in common.

2.2 Training methods

Although improved decoding algorithms can reduce redundant question rates, the underlying issue has not been resolved: the model still assigns a high probability for undesirable response candidates. Several training methods have been proposed to address this problem. For dialogue response generation, (He and Glass, 2019) proposed a negative training framework to resolve the problem of malicious and generic responses. (Welleck et al., 2019) stated that the standard likelihood training objective for text generation is a flawed approach, which contributes significantly to the generation of undesirable behaviors. They then proposed an unlikelihood training objective that forces unlikely generations to be assigned a lower probability by the model. The method is then applied to reduce not only dull and repetitive sentences but also inconsistent and contradictory responses (Li et al., 2021b). Another approach to discourage the model from generating undesirable texts is contrastive training (Cao and Wang, 2021; Li et al., 2022), which aims to differentiate the embedding representations of positive and negative responses.

3 Methodology

3.1 Dialogue generation

The goal of open-domain dialogue generation is to predict the target response $y = (y_1, y_2, .., y_n)$, given the dialogue context $x = (x_1, x_2.., x_m)$ and augmented information $s = (s_1, s_2, .., s_k)$. The dialogue context $x_{1:m}$ is the concatenated history utterances from both speakers while the augmented information $s_{1:k}$ can be scenarios, external knowledge, speaker personas, etc.

Since using the full dialogue context is computationally expensive, prior studies often use a truncated one, e.g. last 128 tokens, alongside personas from both speakers. The introduction of personas is to make sure the newly generated response is consistent with what has been said in the dialogue history. In this study, we propose another utility of speaker personas: to avoid asking redundant questions. For example, if one of the partner's personas is *I am a vegan*, then the chatbot should not ask a question like *What is your favorite kind of meat?*.

To augment the generation with personas, we use the Fusion-in-Decoder (Izacard and Grave, 2020)

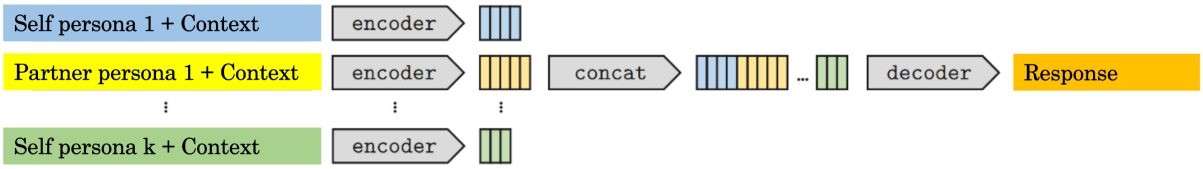

Figure 2: Response generation with augmented speaker personas using Fusion-in-Decoder method.

as shown in Figure 2. We prepend each of the top N personas to the dialogue context and encode them independently using an encoder. The decoder then attends to the concatenated encoding outputs to produce a final response. To extract speaker personas from conversation history, we use a pre-trained BB2 Memory Decoder from ParlAI[2]. All partner personas are used to produce the responses.

3.2 Likelihood training

Given a dataset $D^+ = \{(x^+, s^+, y^+)\}$ collected from real human conversations, we train a response generation model using standard maximum likelihood estimation (MLE)

$$\mathcal{L}_{MLE}(p_\theta, x^+, s^+, y^+) = -\sum_{t=0}^{|y^+|} log\ p_\theta(y_t^+ | x^+, s^+, y_{<t}^+)$$

where x^+ is the truncated dialogue context, s^+ is the speaker personas, y^+ is the next target response, and y_t^+ is the t-th token of y^+.

4 Redundancy mitigation methods

4.1 Unlikelihood training

We apply the unlikelihood loss (UL) (Welleck et al., 2019) to discourage the model from generating undesirable responses. Given an incoherent dataset $D^- = \{(x^-, s^-, y^-)\}$, the loss is computed as:

$$\mathcal{L}_{UL}(p_\theta, x^-, s^-, y^-) = -\sum_{t=0}^{|y^-|} \beta(y_t^-)\ log(1 - p_\theta(y_t^- | x^-, s^-, y_{<t}^-))$$

where y^- is the undesirable response, and s^- contains partner's persona that make y^- a redundant question. $\beta(y_t^-)$ is a candidate-dependent scale that controls how much the token t-th should be penalized. We set $\beta = 0$ for the first two tokens of the question and for tokens that do not belong to the question. The β values for the remaining tokens are set to 1.

[2]https://parl.ai/docs/zoo.html

We train the model with a mixture of likelihood and unlikelihood losses to avoid degradation. The likelihood is performed on D^+ to push up the probability of tokens in the positive response y^+ while unlikelihood is performed on D^- to push down the probability of tokens in the undesirable response y^-. It should be noted that samples from D^+ and D^- can overlap or differ. In this study, we generate D^- using the same samples from D^+.

For each positive sample (x^+, s^+, y^+) in D^+, we generate the corresponding negative one (x^-, s^-, y^-) by keeping x and y: $x^- = x^+$; $y^- = y^+$. We then append an additional partner persona s_{neg} to the existing personas: $s^- = s^+ + s_{neg}$. The negative persona s_{neg} is chosen so that its presence will turn the positive response y^+ into a negative one. For example, if the positive response is *What is your favourite kind of meat?*, then an example of s_{neg} should be *I am a vegan*. A simple strategy to generate s_{neg} is to extract the partner persona from the next response in the dialogue. Figure 3 illustrates how a positive and a negative training sample are generated.

As the samples from D^+ and D^- overlap, the total loss can be now written as follow:

$$\mathcal{L} = \mathcal{L}_{MLE}(p_\theta, x, s^+, y) + \mathcal{L}_{UL}(p_\theta, x, s^-, y)$$

4.2 Classification

As the model can produce multiple responses given the input, we can filter out candidates containing redundant questions. Hence, we can build a binary classification model that can detect whether a generated response contains such questions. The model takes three inputs: the truncated dialogue context, partner speaker persona, and the generated response. Rather than inputting all speaker personas at once for a single prediction, we split them into multiple one-sentence personas and perform multiple predictions. If any of the predictions indicate redundancy in the generated response, we classify it as containing redundant questions.

To generate training data for the classification model, we use the same D^+ and D^- sets discussed

in Section 4.1. For the redundant class, we pair up the negative partner persona s_{neg} with the target response y and dialogue context x. Meanwhile, we replace s_{neg} with a partner persona presented in s^+ to form the non-redundant class.

We fine-tune three pre-trained language models, namely XLnet (Yang et al., 2019), RoBERTa (Liu et al., 2019), and DeBERTa (He et al., 2020), for classification task. Each training sample is formed by concatenating the dialogue context, partner speaker persona, and generated response with a separator token in between.

4.3 Contrastive decoding

To address the repetition problem in text generation, (Su et al., 2022) has proposed a new approach called contrastive decoding. Since the method was originally designed for decoder-only language models (e.g., GPT2), we made some modifications to adapt it to encoder-decoder models.

Given the context x and prefix decoded text $y_{<t}$, the selection of the output token y_t follows:

$$y_t = \arg\max_{v \in V^{(k)}} \{(1-\alpha) \times \overbrace{p_\theta\left(v \mid y_{<t}, x\right)}^{\text{model confidence}} - \alpha \times \underbrace{\max\{sim\left(h_v, h_{x_j^n}\right)\}}_{\text{degeneration penalty}}\}$$

Where $V^{(k)}$ is the set of top-k predictions from the model's probability distribution $p_\theta\left(\cdot \mid y_{<t}\right)$. The representation of token v, denoted as h_v, refers to the decoder output (i.e., the hidden state of the final layer) given the concatenation of the prefix $y_{<t}$ and v, as well as the encoder outputs of the dialogue context x. Similarly, the representation $h_{x_j^n}$ is the decoder output of the j-th token of the n-th turn in the dialogue context. $h_{x_j^n}$ is computed based on the concatenation of the prefix $x^n_{\leq j}$ and x^n_j, as well as the encoder outputs of dialogue context $x^{<n}$. $sim(\cdot,\cdot)$ computes the cosine similarity between token representations while $\alpha \in [0,1]$ controls the importance of model confidence and degeneration penalty. Model confidence refers to the probability assigned by the model to the candidate v, while the degeneration penalty measures the similarity between the candidate v and all tokens presented in the dialogue context. We set α = 0.4 based on the results presented in (Su et al., 2022).

4.4 Contrastive training

Contrastive learning can be used to discourage model from generating undesirable responses (Cao and Wang, 2021). We propose a contrastive training objective that drives the model to favour the generation of non-redundant questions over redundant ones. Given a positive sample $q^+ = (x, s^+, y)$ from D^+ and its corresponding negative sample $q^- = (x, s^-, y)$ from D^-, the objective is to differentiate the question representations between the two samples. Assume that we have a positive set $P = \{q_1^+ = q^+, q_2^+, .., q_m^+\}$ generated from q^+ and a negative set $N = \{q_1^- = q^-, q_2^-, .., q_m^-\}$ generated from q^-, the contrastive loss for q can be written as follow:

$$l = \frac{-1}{\binom{|P|}{2}} \sum_{\substack{q_i^+, q_j^+ \in P \\ q_i^+ \neq q_j^+}} \log \frac{\exp(\text{sim}(\boldsymbol{h}_i^+, \boldsymbol{h}_j^+))}{\sum\limits_{\substack{q_k \in P \cup N \\ q_k \neq q_i^+}} \exp(\text{sim}(\boldsymbol{h}_i^+, \boldsymbol{h}_k))}$$

Where h_i^+ and h_j^+ are representations of q_i^+ and q_j^+, while h_k is representation of q_k, which can be either a sample of the positive or negative set.

Sample construction. Given a positive sample $q^+ = (x, s^+, y)$, we generate its sibling positive/negative samples by keeping x and y but appending an additional partner persona s_{add} to the existing personas s^+. s_{add} is chosen from a persona pool S, which is a collection of all speaker personas extracted from the training set. First, we rank personas in S based on their similarity scores to the context x and then pick the top-k personas as s_{add}. After that, we use the redundant classifier from Section 4.2 to classify the each input (x, s_{add}, y). If the prediction is redundant, we use s_{add} to generate a negative sample, otherwise we use it to construct a positive one.

Sample representation (h_*). We use the outputs of the decoder's last layer to form the representation h for each positive and negative sample. More specifically, we only average over tokens that belong to the question in the target response y.

Training. To avoid model degradation, we combine contrastive loss with the original MLE loss $\mathcal{L} = \mathcal{L}_{MLE} + \mathcal{L}_{CL}$.

4.5 Unlikelihood training with augmented loss

We reuse the sample construction method from Section 4.4 to increase the coverage of the training set and boost the performance of unlikelihood training.

More specifically, we augment the original unlikelihood loss with loss computed from sibling positive and negative samples as follow:

$$\mathcal{L}_{aug} = \frac{1}{|P|}\sum_{i=1}^{|P|}\mathcal{L}_{MLE}(p_\theta, x, s_i^+, y) + \frac{1}{|N|}\sum_{j=1}^{|N|}\mathcal{L}_{UL}(p_\theta, x, s_j^-, y)$$

Where P and N are the positive and negative sets. s_i^+ is the speaker persona of i-th sample from P and s_j^- is the speaker persona of j-th sample from N. Samples from P and N are included in the same batch of training. Using augmented loss helps the model better distinguish between negative and positive samples, which reduces the number of redundant questions while maintaining quality of the original model.

5 Experiments setup

5.1 NRQ dataset

As there is no available dataset addressing the problem of redundant questions, we create a new non-redundant question set called NRQ, which consists of positive training samples for D^+ and negative samples for D^-. To form our D^+, we gather training samples from Wizard of Wikipedia (WoW) (Dinan et al., 2018), Empathetic Dialogues (ED) (Rashkin et al., 2018), Blended Skill Talk (BST) (Smith et al., 2020), Multi-Session Chat (MSC) (Xu et al., 2021), and Wizard of Internet (WOI) (Komeili et al., 2021) datasets. Note that we only select samples with questions presented in the target response. To extract speaker personas from conversation history, we use a pre-trained Dialogue Summarization Model from ParlAI.

To create negative samples for the NRQ dataset, we use the approach outlined in Section 4.1, illustrated in Figure 3. Specifically, we convert each positive sample (x, s^+, y) into a negative one by augmenting the speaker personas s^+ with a negative partner persona s_{neg} (e.g. *I have two girls*), which we obtain from the partner personas of the next dialogue turn (e.g. *Yes, I have two girls*), denoted as s_{next}. However, this procedure poses two challenges: (i) s_{next} may contain multiple personas, some may not be relevant to the questions posed in the target response y, (ii) s_{next} may be entirely irrelevant, for instance if the next dialogue turn is off-topic or the persona extractor model

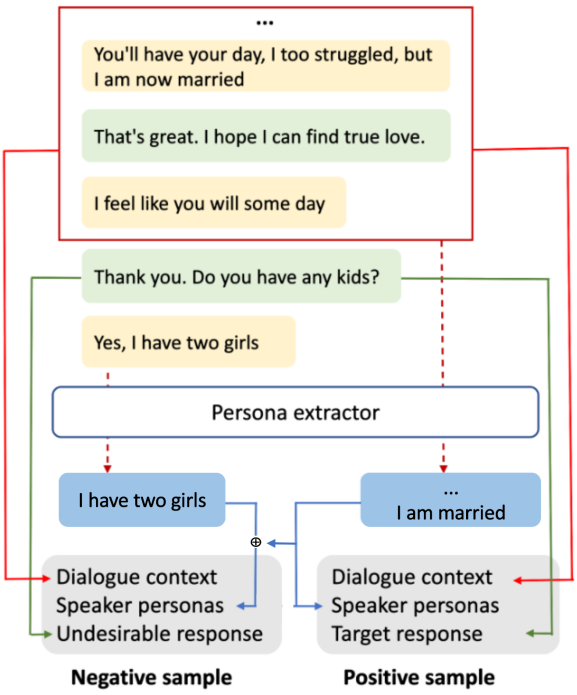

Figure 3: A training sample of NRQ dataset

fails to identify the correct personas. As a result, we rely on human annotators to select only the relevant s_{neg} from s_{next} and discard samples where no relevant s_{neg} can be found. The number of samples in NRQ is 100,181 before filtering, and 50,178 after filtering. We split the final dataset into 46,286 for training, 2,000 for validation, and 1,892 for testing.

Redundant question classification. As described in Section 4.2, we use D^+ and D^- to generate training data for our the redundant question classifier, resulting a total of 48,297 and 45,494 samples for redundant and non-redundant class respectively. In addition, we incorporate human annotation results mentioned above where the negative persona s_{neg} is deemed irrelevant to the question. This provides an additional 39,271 non-redundant samples.

5.2 BB2 Baseline

As training an end-to-end generation model from scratch is computationally expensive, we choose to use the pre-trained BB2 model (3 billion parameters) as baseline. Our goal is to reduce the number of redundant questions generated by the model. The BB2 model is fine-tuned from the Blenderbot1 model (Roller et al., 2020b) on BST, MSC, and WOI datasets. For decoding, we use beam search with 4-gram blocking to prevent repetitive questions from generating. The maximum number of tokens in the dialog context is set to 128.

5.3 Evaluation

Perplexity (PPL) is a metric to measure how well a generation model predicts a response. We want the model to output low perplexity scores for good and coherent responses while producing high perplexity scores for undesirable responses such as redundant questions in our case.

Diversity measures lexical diversity of generated texts, which is computed based on corpus-level repetition at different n-gram levels as follow: **diversity** $= \prod_{n=2}^{4}(1.0 - \frac{rep-n}{100})$, where **rep-n** $= 1.0 - \frac{|unique\ n-grams(C)|}{|total\ n-grams(C)|}$; C is a collections of generated responses by the model.

Coherence measures the semantic similarity between dialogue context and generated response. We use SimCSE following (Su et al., 2022) to compute the similarity in the embedding space.

Redundant question rate is the percentage of generated questions that are redundant. For automatic evaluation, we use the classifier presented in Section 4.2 to check if a question is redundant.

Automatic evaluation is essential for hyperparameter tuning and model selection. To automatically estimate quality of generated texts, we first perform self-chat, i.e two chatbots chatting with each other, to generate 50 bot-bot dialogues using BB2 Baseline. To make sure each dialogue is different, we seed each one with a human-human conversation (25 turns) from the MSC Session1&2 and then generate 40 more turns. After that, we calculate diversity, coherence, and redundant rate scores based on the generated questions.

Human evaluation. We recruited human annotators from Amazon Mechanical Turk to conduct 50 human-bot conversations for evaluation. We seed each human-bot conversation with 25 turns from MSC Session1&2. The human and the bot, i.e BB2 Baseline, are asked to continue each seeded conversation for 40 turns. After that, we asked another group of annotators to manually check if each generated question is a redundant question based on the entire conversation.

Method comparison. We propose a method for a fair comparison between the BB2 Baseline and other approaches mentioned in Section 4. Instead of having each model conduct its own conversations, we use responses generated by the BB2 Baseline as a ground for comparison. For each of the BB2-generated questions, we regenerate it with the compared models and then recompute the evaluation scores. In cases where a model does not generate any questions at the end, we replace the end-of-sentence token with the most probable question-words token (e.g. what, how, when, etc) and continue the decoding process.

Models	Acc	F1-score	
		Redundant	Non-redundant
XLNet	88.3%	85.9	90.0
RoBERTa	88.6%	86.3	90.1
DeBERTa	88.2%	86.5	89.5

Table 1: Redundant question classification results on the test set. *Acc* stands for accuracy.

5.4 Training configuration

We fine-tune the BB2 Baseline using one A100 GPU with an Adam optimizer. The learning rate and batch size are set to 5e-6 and 8. The model is fine-tuned in a multi-task fashion using samples from BST, MSC, WOI, and NRQ datasets. We draw samples from each task equally in a round-robin fashion. We use early stopping based on the combined score of test set perplexity and redundant question rate of bot-bot conversations.

6 Experiment results

Redundant question classification. We first report performances of our redundant question classifier in Table 1. As can be seen, all three models perform similarly well, with RoBERTa achieving the highest accuracy of 88.6%. Therefore, we choose RoBERTa to automatically calculate the redundant question rate of the generation models in subsequent analyses.

Conversation length vs redundant rate. As shown in Figure 4, the redundant question rate increases significantly with respect to the length of the conversation. For BB2 Baseline, the rate is 18.4% at turn 30. The number further increases by another 8.1% when the conversation reaches 65 turns. However, this issue is not a concern in previous studies as most evaluate the chatbots on a short conversation setting (less than 10 turns). The increase in redundant rate can be attributed to the limited number of topics the chatbot can initiate. When the conversation is prolonged, it often revisit topics that have already been discussed.

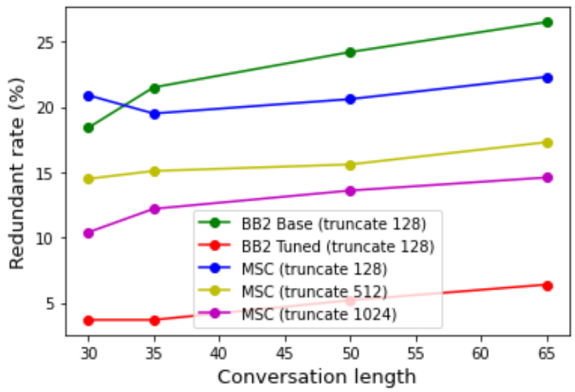

Figure 4: The impact of conversation length and truncated context length on redundant question rate.

Most common redundant questions
Do you have any pets?
What kind of dog do you have?
What do you do for a living?
What are you studying in school?
What kind of music do you like?
Most common questions in training data
What do you do for a living?
Do you have any pets?
Do you have any hobbies?
Where are you from?
What music do you like?

Table 2: Most common redundant questions generated by the BB2 Baseline and most frequent questions presented in the training data of the model.

Truncated context length vs redundant rate. The limitation of 128 tokens for truncated dialogue in the BB2 Baseline could be the cause of higher redundant question rate. Increasing the truncation length could be considered as a possible solution to address this issue. To investigate this hypothesis, we utilized the MSC model (Xu et al., 2021), which was specifically trained on the MSC dataset to effectively handle long conversations. In Figure 4, the results demonstrate a significant reduction in redundancy rates by extending the truncation length. For conversations with a length of 30, the redundancy rate decreased from 18.4% (truncated at 128) to 10.4% (truncated at 1024). However, it is important to note that despite these improvements, they still fall short compared to the BB2 Tuned model using our proposed methods, while also incurring increased training and inference costs.

Bias in training data. Another contributing factor to the redundant issue is the bias of the BB2 Baseline towards common topics, such as pets, hobbies, and careers, which increases the likelihood of repeating the same topics over again. An explanation can be seen in Table 2, which shows the most frequent redundant questions generated by the BB2 Baseline. Obviously, these questions strongly overlap with the most frequent questions in the training data of BB2 Baseline, demonstrating the model's tendency to generate the most probable questions as a downside of maximum likelihood estimation.

Mitigation methods comparison. We apply mitigation methods to improve the performance of BB2 Baseline. As can be seen in Table 3, our proposed methods are able to not only reduce the redundancy rate but also increase the diversity score. Discussions for each method is provided as below:

BB2 Baseline does not perform well in most metrics. The negative perplexity is significantly lower than the positive one, indicating that the model is more likely to generate redundant questions instead of target questions. Additionally, the low measure of lexical diversity suggests that the model tends to produce common but repetitive questions, resulting in a high redundant rate of 26.5%.

Contrastive decoding can significantly reduce the redundant question rate to 17% without the need to retrain the model. This improvement can be explained by the significant increase in diversity score, indicating that the model favors less repetitive questions. We also observe an improvement in coherence score, which is consistent with prior studies (Su et al., 2022).

Unlikelihood training obtains the best redundant rate at 7.5%, thanks to significant increases in negative PPL and diversity score. The slight increase in positive PPL suggests a tiny degradation in the quality of the generated questions, which demonstrates by a lower coherence score. However, using augmented loss and further combining with contrastive decoding bring considerable improvements across all metrics, especially in diversity score.

Contrastive training reduces the redundant rate to 11.4% but it is still pales in comparison to unlikelihood training. Also, using contrastive training comes at the cost of question degeneration, as demonstrated by the increase in both negative and positive PPL. It can be seen that the model is confused between the task of degenerating redundant questions versus degenerating all questions.

Methods	Positive PPL	Negative PPL	Coherence	Diversity	Redundant rate
BB2 Baseline	12.2	7.9	0.34	0.02	26.5%
Contrastive decoding	-	-	0.36	0.07	17.0%
Contrastive training	14.4	69.6	0.34	0.11	11.4%
Unlikelihood training	12.5	37.5	0.32	0.09	7.50%
+ Augmented loss	12.7	38.0	0.33	0.12	6.44%
+ Contrastive decoding	-	-	0.33	0.15	6.66%

Table 3: Performances of different redundancy mitigation methods. Positive PPL refers to the perplexity of target questions from positive samples, while negative PPL refers to the perplexity of redundant questions from negative samples. We compute the positive PPL on the combined test set of BST, MSC, WOI, and NRQ. Negative PPL is computed on the NRQ test set. Coherence, diversity, and redundant rate are computed on the generated questions from 50 bot-bot conversations.

Methods	Redundant
BB2 Baseline	27.2%
Classification	15.4%
Unlikelihood	11.4%
Unlikelihood + Classification	8.7%

Table 4: Evaluation results on 50 human-bot dialogues

BB2 Baseline	BB2 Tuned
37.8%	62.1%

Table 5: Win rate of the BB2 Baseline and our proposed approach.

Human evaluation. Table 4 reports human evaluation results on 50 human-bot dialogues. The results indicate that the BB2 Baseline still has a high redundant question rate of 27.2%, highlighting the need for effective solutions. While using a redundant classifier alone can reduce the rate significantly to 15.4%, this is still much higher than the 11.4% rate achieved with unlikelihood training. The failure of the redundant classifier can be attributed to two reasons: (1) Since the problem of assigning high probabilities to redundant questions remains unaddressed, it is not uncommon that the model generates all candidate responses with redundant questions (2) With an accuracy of 88.6%, the redundant classifier can misclassify some redundant questions as non-redundant. Nevertheless, using classification on top of unlikelihood training can reduce the redundant rate further to 8.7%.

We can see that the improvements in human-bot conversations are considerably lower compared to bot-bot conversations. This is due to the fact that human-bot conversations are typically more varied and less predictable than bot-bot conversations. In contrast, bot-bot conversations tend to revolve around common topics and employ a shared vocabulary that is well-represented in the training data of the NRQ dataset.

Finally, we asked human annotators to compare the overall question-asking ability of the original BB2 Baseline with our proposed method combing unlikelihood training with redundant classifier. For each pair of comparisons, two annotators were asked to choose which of the two generated responses was better, or if they were both equally good or bad. In cases where the annotators disagreed, we manually reviewed the case and determined the correct annotation. When calculating the win rate, we excluded comparison cases where both responses were equal in quality. According to the results presented in Table 5, our approach significantly outperforms the original model.

7 Predictions analysis

We present several successful and failed cases of the proposed approach. Table 6 compares perplexities of the BB2 Baseline and BB2 tuned with unlikelihood training in generating the target questions based on different partners' personas. On the one hand, if the partner's persona, i.e *I have a dog*, has nothing to do with the target question, i.e *What do you do for a living*, then there is not much difference in perplexity between BB2 Baseline and BB2 Tuned. This suggests that the proposed negative training method does not badly affect the question-asking ability of the original BB2 Baseline. On the other hand, if the presence of the partner's persona, i.e *I'm a software engineer*, turns the target question into a redundant question, then the perplexity of the BB2 Tuned model increases significantly to 68.5 while the number for BB2 Baseline remains

Questions	Partner's persona	Question perplexity	
		Baseline	Tuned
What do you do for a living?	I have a dog.	2.04	2.42
	I'm a software engineer.	2.06	68.5
	I'm still in high school.	2.07	3.41
Do you have any pets?	I like to read books.	2.56	2.49
	I have a cat and a dog.	2.52	50.0
	My apartment doesn't allow pets.	2.48	2.93

Table 6: Example perplexities of the BB2 Baseline and BB2 Tuned with NRQ when predicting the target questions.

very low, at 2.06. We also note that one of the weaknesses of the BB2 Tuned model is that it is still unable to spot redundant questions if they are not clearly related to the partner's persona. For instance, the partner's persona *I'm still in high school* can be interpreted as *I don't have a job* but the BB2 Tuned model still assigns a very low perplexity for the redundant question *What do you do for a living*.

8 Conclusion

Asking good questions is an important skill for a chatbot to engage in a long-term conversation. This study first introduces the problem of redundant questions in neural text generation models. Several methods, including negative training, decoding, and classification have been proposed to lower the probabilities of these undesirable questions. We also create the first-of-its-kind dataset named NRQ dataset containing training samples with a redundant question assigned to each dialogue context and speaker personas. We validate our methods with the BB2 model and observed a significant reduction of the redundant rate, which results in a higher rating for the questioning skills of the chatbot. We believe the proposed approaches and datasets will be beneficial for building future dialogue systems.

9 Acknowledgement

This work was funded by Science Foundation Ireland through the SFI Centre for Research Training in Machine Learning (18/CRT/6183).

Limitations

Resource hungry. One of the difficulties in deploying large-scale neural text generation models is resource allocation and latency problems. For example, the BB2 Baseline 3B requires at least a 16GB GPU and a couple of seconds to generate the response using one Tesla V100. As our approach requires inputting all of the partner's persona alongside dialog context, it almost doubles the inference time and increases the use of GPU memory significantly. As a result, it is not resource-friendly when the conversation is prolonged. A possible solution to this is to use the RAG retriever model to select a few relevant partner personas and incorporate only these into the input. However, this may be difficult to do so as we might not know what questions are going to be generated during decoding. A redundant question might be generated because a partner's persona is missing.

The redundant rate is still high. Although the proposed approach significantly reduces the redundant question rate, the number still remained relatively high, at 8.7%. We believe this is a much more serious issue compared to other challenges, such as contradiction or "hallucinations", as it is very uncomfortable for the user to repeat the same information or discuss a topic multiple times during the conversation. As mentioned in the previous sections, one of the main weaknesses of the fine-tuned model is the failure in recognizing the indirect relations between a speaker persona and a redundant question. We believe the problem can be addressed by scaling up the size of the NRQ dataset to cover more of these difficult cases. Better data augmentation techniques can also be used to diversify redundant questions and negative partner personas.

References

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.

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in

abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Moya Chen, Douwe Kiela, Mojtaba Komeili, Spencer Poff, Stephen Roller, Kurt Shuster, Arthur Szlam, Jason Weston, and Jing Xu. 2021. Blender bot 2.0: An open source chatbot that builds long-term memory and searches the internet. https://parl.ai/projects/blenderbot2/.

Emily Dinan, Gavin Abercrombie, Stevie A Bergman, Shannon Spruit, Dirk Hovy, Y-Lan Boureau, Verena Rieser, et al. 2022. Safetykit: First aid for measuring safety in open-domain conversational systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Dilek Hakkani-Tur. 2021. Alexa prize socialbot grand challenge year iv. In *Alexa Prize SocialBot Grand Challenge 4 Proceedings*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Tianxing He and James Glass. 2019. Negative training for neural dialogue response generation. *arXiv preprint arXiv:1903.02134*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

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

Jakub Konrád, Jan Pichl, Petr Marek, Petr Lorenc, Van Duy Ta, Ondřej Kobza, Lenka Hỳlová, and Jan Šedivỳ. 2021. Alquist 4.0: Towards social intelligence using generative models and dialogue personalization. *arXiv preprint arXiv:2109.07968*.

Ilya Kulikov, Alexander H Miller, Kyunghyun Cho, and Jason Weston. 2018. Importance of a search strategy in neural dialogue modelling. *arXiv preprint arXiv:1811.00907*.

Weizhao Li, Junsheng Kong, Ben Liao, and Yi Cai. 2022. Mitigating contradictions in dialogue based on contrastive learning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2781–2788.

Zekang Li, Jinchao Zhang, Zhengcong Fei, Yang Feng, and Jie Zhou. 2021a. Addressing inquiries about history: An efficient and practical framework for evaluating open-domain chatbot consistency. *arXiv preprint arXiv:2106.02228*.

Zekang Li, Jinchao Zhang, Zhengcong Fei, Yang Feng, and Jie Zhou. 2021b. Addressing inquiries about history: An efficient and practical framework for evaluating open-domain chatbot consistency. *arXiv preprint arXiv:2106.02228*.

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

Ashwin Paranjape, Abigail See, Kathleen Kenealy, Haojun Li, Amelia Hardy, Peng Qi, Kaushik Ram Sadagopan, Nguyet Minh Phu, Dilara Soylu, and Christopher D Manning. 2020. Neural generation meets real people: Towards emotionally engaging mixed-initiative conversations. *arXiv preprint arXiv:2008.12348*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2018. Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv preprint arXiv:1811.00207*.

Stephen Roller, Y-Lan Boureau, Jason Weston, Antoine Bordes, Emily Dinan, Angela Fan, David Gunning, Da Ju, Margaret Li, Spencer Poff, et al. 2020a. Open-domain conversational agents: Current progress, open problems, and future directions. *arXiv preprint arXiv:2006.12442*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020b. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

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

Eric Michael 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. *arXiv preprint arXiv:2004.08449*.

Yixuan Su and Nigel Collier. 2022. Contrastive search is what you need for neural text generation. *arXiv preprint arXiv:2210.14140*.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *arXiv preprint arXiv:2202.06417*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. 2022. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. *arXiv preprint arXiv:2206.02369*.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.