Engagement-based response generation for open-domain dialogue

Stanford CS224N Custom Project

Marcelo Peña
Department of Computer Science
Stanford University
marcelop@stanford.edu

Ernesto Nam
Department of Computer Science
Stanford University
enamsong@stanford.edu

Abstract

Existing open-domain dialog models are trained to minimize the perplexity of target human responses. However, some replies are more engaging than other and they generate more followup interactions. We finetune GPT-2 using Reddit data on comment upvotes and engagement, so that, given an input sentence and context, we generate a response optimized for human engagement. We found that by fine-tuning GPT-2 on filtered dataset containing only high-engagement responses (70th percentile), and including a discrete representation of the sampled engagement rank, we can generate an response that is 15% more engaging than the human-generated benchmark in average.

1 Key Information to include

- Mentor: Lisa Li (xlisali@stanford.edu)
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Our task is to create a model that, given an input sentence and context, can return a more engaging response to the context. This NLP task, called response generation, can be useful for politicians / public interest entities that seek to engage with their audiences on social media and drive as much engagement as possible at scale. Moreover, it can be helpful in keeping reader's interest on human-computer interaction applications, such as chatbots.

This is a historically challenging problem is challenging: conventional quality measurements such as reference-based similarity (Papineni et al., 2002) or lexical diversity (Li et al., 2016; Zhang et al., 2018b) capture only limited aspects of response quality, and are not strongly predictive of human reactions: simply because a response is different from others does not necessarily mean that it will be perceived as "bad". They don't represent language fully because they fail to grasp its social aspect. So engagement optimization is very challenging since we can't use data labeling to evaluate models on engagingness: labeling the "engagingness" of a response is not something an annotator can do, it's a more subconscious response to an answer.

Instead, we use Reddit as a large-scale, collective vote on what responses are more / less engaging. A similar approach was used in training DialogRPT, a engagement classifier, and our extension into text generation can be promising in its applications to other tasks (marketing, sales, political campaigning, copywriting, etc.)

3 Related Work

DialogRPT DialogRPT is a set of transformer-based models trained on 133M pairs of human feedback data developed by Microsoft Research in 2020. It serves a a ranker of (context, comment) pairs based on their expected compellingness. Particularly, this ranker is currently the main method to predict comment compellingness, as it largely outperforms the conventional dialog perplexity baseline on predicting Reddit feedback, and has a strong performance on zero-shot tests predicting compellingness on other platforms, such as Facebook and Twitter. This is because the ranker was trained to reduce the cross entropy between the target distribution of contrastive samples (c, r^+) and (c, r^-). Through crowd-sourced human evaluation, Microsoft research shows that this contrastive learning approach for ranking correlated better with real human preferences than other baseline models. We will use DialogRPT to rank the compellingness of our comment generations across their three rankers.

Quark: Controllable Text Generation with Reinforced Unlearning

The main paper we based off of in terms of methodology is Quark: Controllable Text Generation with Reinforced Unlearning by Yejin Choi, et al. In this paper, the authors propose the Quantized Reward Konditioning (Quark) algorithm, which optimizes a reward function by quantifying both desired and undesired features. The algorithm proposed in the paper specifically sorts the samples collected in quantiles that contain a unique reward and uses a standard language modeling loss on the samples from each quantile and based on their tokens. We will use a similar approach for pre-processing our data for finetuning. During the actual text-generation, the algorithm conditions on a high-reward token, hence gaining greater control on the emphasis of the words generated as well as avoided. As an example, this process helps exhibiting less unwanted properties such as words associated with toxicity or negative sentiment.

A Systematic Evaluation of Response Selection for Open Domain Dialogue

While the paper about Quark shed light on us on fine-tuning ideas, we also relied on the paper "A Systematic Evaluation of Response Selection for Open Domain Dialogue" by Amazon Alexa researchers to understand existing work on open domain dialogue generation. Open dialogue is also a topic emphasized in the paper of DialogRPT, given its higher complexity due to the open-endedness of topics as well as importance because of its tendency of being more realistic. This research by the Amazon Alexa researches examines the challenge of building effective open-domain chatbots, and proposes a solution of generating multiple response candidates and selecting the best one. Previous research has focused on training response rankers using synthetic data, which constructs inappropriate responses using random selection or adversarial methods. However, in this work, the authors curated a dataset where a good number responses from multiple generators were manually annotated as appropriate or inappropriate. They researchers argue that this training data better matches actual use cases, enabling models to rank responses more effectively. The authors demonstrate that using both multiple positive candidates and manually verified hard negative candidates can improve performance compared to using adversarial training data.

4 Data

We utilized the Reddit datasets from pushshift.io made available by fhoffa on Bigquery. As context, Reddit is a social media site with a forum-style discussion structure, where users create posts in topic-based communities called subreddits and interact by commenting in threads. In turn, Pushshift.io is a public data-storage that uses Reddit API to offer data about Reddit posts and comments. DialogRPT was trained on datasets provided by this same source, specifically with the Reddit Posts and Reddit Comments dataset from Jan, 2012 to Dec, 2013. A Reddit comment entry in the database contains the post's titles and selftext (elaboration on the post's title), while Reddit comment has the body of the comment and the id of the thread it belongs to, which can be a post or another comment and is a an important structure to take into consideration to run the DialogRPT model.

For our data pre-processing, we accessed the datasets of Reddit posts and Reddit comments to then retrieve their DialogRPT score and, subsequently, train our GPT-2 text-generation model on the output. Specifically, we utilized the datasets from January 2016 to January 2018, a timeframe that was not used to train DialogRPT (as per their paper). We selected 130k distinct entries for pretraining and 5k for testing. The datasets also involved rigorous cleaning before running the model. In particular, we

followed a similar cleaning strategy employed by DialogRPT, adding some particular considerations to ensure that the entries could be processed by GPT-2:

- Filter by comments with length of less than 512 characters to maintain emphasis on dialogue instead of long text. This will respect the chat GPT-2 usage limits.
- Mask URL's in posts and comments were by the string '(URL),'
- Pair comments with their respective parent posts or threads across distinct datasets in the format of [context, comment1, comment2].

The dataset sizes after cleaning comprise as follows:

- Jan, 2016 - Dec, 2016: 52k entries

- Jan, 2017 - Dec, 2017: 80k entries

- Total: 130k entries

5 Approach

Starting from a pretrained language model (GPT-2), we ran five steps. We explain them in detail below.

1. **Exploration & data pre-processing:** We retrieved 500k pairs of posts and comments using the Reddit API. Although the data we required is similar to the training data used by DialogRPT, their paper listed API endpoints that were no longer available. As a result, we pre-processed the raw comment data with our own python script (included in file XXX, more on this below on "Data"), tokenized it and truncated to 1024 tokens to make it compatible with GPT-2 usage limits. We then formed context-comment pairs from posts, and recorded three key measures for each comment: updown (number of upvotes), depth (number of subsequent comments), and width (average length of subsequent comments).
2. **Quantization:** DialogRPT returns a normalized rank of comment compellingness $r(x, y)$ across each of the three key measures. Here $x = (x_1, \ldots, x_|x|)$ and $y = (y_1, \ldots, y_|y|)$ are sequences of tokens corresponding to the context and the corresponding comment. So, for all context-comment pairs, we evaluated its reward by using the DialogRPT model, and built a Datapool D_0 of tuples $(x, y, r(x, y))$. We then sort D_0 based on decreasing $r(x, y)$ values and partition the sorted pool into 3 equally sized quantiles - D^1, D^2, D^3 - corresponding to their level of engagement - low, medium, high. Each sample (x, y) is now part of a quantile that is identified by a discrete rank token $r_k, k \in 1, 2, 3$, a continuous rank token $r_k, k \in [0, 1]$ and a corresponding label. For example, for the context "I love NLP!", the response "Me too!" would be ranked in D^1 (low compellingness) and contain token r_1, while the response "Great! This (URL) has a great resource to learn NLP." is ranked in D^3 (high compellingness), identified by token r_3.
3. **Supervised Finetuning:** For the finetuning step, we trained on the quantized datapool using a standard conditional language modelling objective - maximizing likelihood:

$$max_\theta E_{k\sim\mu(1,K)} E_{(x,y)\sim D^k} [\log p_\theta(y|x, r_k)]$$

4. **Generation:** For generation, we use greedy search, which selects the word with the highest probability as its next word according to:

$$arg\, max_w P(w|w_{1:t-1})$$

at each step t. We will obtain the conditional probabilities from the GPT-2 model p_θ.

The format of the prompts generated is the following:

<| RedditComment |> {context} {token} {comment} <| endofline |> Where the token can be an int representing the compellingness rank (0 - 10), a float [0.00 - 1.00], or a string ('high', 'medium, ' low').

Moreover, after a batch of comments are generated, we filter for complete sentences only (delimited by a period, exclamation mark or question mark).

5. **Evaluation:** To evaluate the generated comments, we use the following metrics:

(a) Compellingness rank, as evaluated by running DialogRPT on the generated comment

(b) Comment perplexity, defines as the inverse probability of the test set, normalized by number of words.

$$PP(W) = P(w_1 w_2 ... w_N)^{-\frac{1}{N}}$$

Where $w_1 ... w_N$ are the words in a comment and N is the total number of words in the comment.

(c) Lexical diversity of the comment, as the % of unique N-grams in the comment (or distinct words in the string).

In addition, we compare our results against the following benchmarks:

(a) Average rank of human-generated Reddit comments: by upvote count, response comment width, response depth, and the average of the three

(b) Average perplexity of human-generated Reddit comments

(c) Average lexical diversity of human-generated Reddit comments

6 Experiments

6.1 Evaluation method

DialogRPT scores of generated text Our main evaluation was the DialogRPT scores of our text generated with our fine-tuned GPT-2 model. After obtaining the text generated with respect to a given context through the different fine-tuned methodologies explained in detail below, we got the scores of those texts and compared them to the average scores the original comments had.

Fluency metrics: perplexity and lexical diversity We used fluency metrics to evaluate the cohesiveness of our generated text. We calculated perplexity For lexical diversity, we also took advantage of the NLTK library to tokenize all of our generated text. After the tokenization, we calculated the percentage of unique n-grams of each batch of generated text.

6.2 Experimental details

We ran a set of 6 main experiments, varying both the domain of the dataset to be used in the finetuning process (full dataset of comments vs. filtered by high responses only), and the labels to be used in the prompt supplied to GPT-2. These experiments are summarized in the table below.

Finetuning	Criteria	Token	Experiment #
Full dataset	Updown	Discrete (int)	3
		Continuous (float)	2
		Label ("high", "medium", "low")	1
	Total	Discrete (int)	5
High compellingness only (70th percentile)	Updown	Discrete (int)	4
	Total	Discrete (int)	6

Every experiment used the following parameters to finetune the GPT-2 model:

- **Number of samples:** 50k
- **Max entry length:** 1024 tokens or 35 words
- **Top_p:** 0.8
- **temperature:** 1.

Moreover, all experiments used a test set of 1000 (context, comment) pairs and generated one high-engagement comment per entry. This allowed us to have statistically-significant results. Moreover, we implemented simple gradient accumulation in our pretraining to optimize the process since GPT-2 is large and in our initial tests, the pretraining phase with 20k entries took several hours. Instead, using gradient accumulation we reduced this to around 1h30 for each experiment. We ran all experiments on the Google Colab platform, which are typically NVIDIA V100 or A100 Tensor Core GPUs.

7 Results and analysis

- Training our model on classified samples, meaning only samples that had scores of "highly engaging" proved to have higher overall results of about 5.8 in DialogRPT scores than training the model on all samples. We had hoped that the model would learn what makes a comment good, medium or bad, but in reality this lead us to mixed results. Rather, we found better outcomes when training only on high quality comments. We believe this is because it is hard to prompt the model to generate a good comment vs. a bad comment. In our generation step, we include a token in the context (<|high|> or <7.0>) to indicate that we want to generate a high engagement comment. However, this approach didn't prove to be as effective at filtering out less engaging generations when training on the entire dataset.

Experiment	Updown Mean	Depth Mean	Width Mean	Final Mean
Dataset	0.548224	0.552977	0.582555	0.561252
Experiment 1	0.404288	0.493413	0.530671	0.476124
Experiment 2	0.582203	0.613900	0.653218	0.592109
Experiment 3	0.616060	0.664514	0.704969	0.661848

Figure 1: Mean DialogRPT scores for experiments 1-3

Experiment	Updown Mean	Depth Mean	Width Mean	Final Mean
Dataset	0.672	0.592	0.631	0.632
Experiment 4	0.703	0.715	0.759	0.726
Experiment 5	0.682	0.614	0.687	0.661

Figure 2: Mean DialogRPT scores for experiments 1-3

- Using a discrete token (int from 0 - 10) in the training prompt leads an average 12% to better results than using a continuous token (float from 0 - 1) or a string label ('low', 'medium', 'high'). We initially used continuous token from 0-1 to label our data because it was the format outputted by DialogRPT. However, when checking both the overall results and fluency metrics in either cases, we had better results when using discrete, integer tokens, as represented in Figures 1 and 2.
- Filtering by updown label is 15% a better predictor of compellingness as measured by DialogRPT than using an average of updown, depth, and width. We believe this is because upvoting is a simpler task than commenting, and so it reflects a wider range of human reactions.
- In virtually all cases, text generated with our fine-tuned model scored better when it comes to perplexity and lexical diversity. Specifically, the lexical diversity score were consistently higher than the original comments in all experiments and the perplexity score were lower, except for experiment 1, where the score was 0.01 higher than the original comments. We were particularly surprised by how experiment 4 was around ten percent higher than the original comment.

Experiment	Lexical Diversity	Perplexity
Dataset	69.950000	0.230000
Experiment 1	72.870000	0.240000
Experiment 2	71.660000	0.210000
Experiment 3	74.159472	0.170212

Figure 3: Fluency scores for experiments 1-3

Experiment	Lexical Diversity	Perplexity
Dataset	66.32	0.24
Experiment 4	73.55	0.21
Experiment 5	68.44	0.20

Figure 4: Fluency scores for experiments 4-5

8 Conclusion

We developed an engagement-based response generator for open-domain dialogue by finetuning GPT-2 using Reddit data on comment upvotes and engagement, and evaluating these results on DialogRPT. We found that by fine-tuning GPT-2 on filtered dataset containing only high-engagement responses (70th percentile), and including a discrete representation of the sampled engagement rank, we can generate an response that is 15% more engaging than the human-generated benchmark in average.

The primary limitation of the work is that, although a significant portion of the comments are coherent and engaging, upon visual inspection, some of the generated comments are sometimes nonsensical. Moreover, there is a moral limitation to our work: the comments might reflect the inherent biases of the dataset they're finetuned in. In this case, since they're trained on Reddit data, we observed that some generated comments had hate speech and even transphobic content. Thus, further filtering is needed to avoid perpetuating these biases in the generated interations.

For future work, there are two main avenues to explore. First, we would like to finetune our model with a larger dataset, training and testing against other sources apart from Reddit data (such as Twitter / Facebook). However, due to compute and time constraints, this was discarded for this version of the project. Second, it would be interesting to apply a reinforcement learning approach to this problem, and benchmark this against our supervised finetuning approach.

References