CP2R: GPT-2 Conversational Pipeline using Relevance and Realism Discrimination

Ryan Cooper
College of Computing
ryan.cooper@gatech.edu

Abhishek Mallemadugula
College of Computing
abhishekmalle@gatech.edu

Christian Boylston
College of Computing
cboylston@gatech.edu

Jonathan Leo
College of Computing
jpleo122@gatech.edu

Abstract

GPT-2 and it's successor GPT-3 have taken the open-text generation domain by storm having been leveraged in many open-domain conversational agents. Given their black-box approach to text generation, these models can at times produce inappropriate or unkind responses to sensitive topics. We present CP2R, a GPT-2 based pipeline that attempts to overcome these pitfalls through a fine-tuning over 16 million posts and comments from the CasualConversation subreddit, where comments and posts are explicitly moderated for niceness. Our model generates candidate responses for a given text and picks the most appropriate and realistic one through a combination of outputs from our BERT-based realism discriminator and relevance discriminator.

1. Introduction

In this work, we introduce CP2R (**C**onversation **P**ipeline using **R**elevance and **R**ealism), a GPT-2 pipeline to generate kind responses to a given prompt. The pipeline consists of a model fine-tuned to posts and replies from the CasualConversation subreddit and a post-processing pipeline to select for realistic, relevant responses. Current GPT models are being deployed in a variety of academic and commercial applications where the unpredictable responses of GPT could potentially do harm. We aim by fine-tuning our models to text from a domain moderated to require friendly content, we can construct a model that will mitigate the potential for antagonistic responses.

1.1. Background and Motivation

In the years since their inception, GPT-2 [9] and it's successor GPT-3 [1] have established themselves as the state of the art in open text generation. These models have found many commercial uses in tasks such as open-domain conversation [7], email generation [2], and data augmentation [5]. In recent months, the commercial applications of this model have come under fire after a conversational agent utilizing GPT-3 in the medical domain told a test patient to kill themselves [10]. Recent works have found success in training transformer language models over text that has been moderated specifically for a given task like Wikipedia edit data for the stance neutralization [8]. We follow a similar protocol training our GPT-2 language model over text from the CasualConversation subreddit. CasualConversation [1] is a subreddit that defines itself as the "The friendlier part of Reddit" requiring that users be very respectful of others and refrain from any offensive behavior. We hope our pipeline can be used for further fine-tuning in different domains to help generate kinder, more sensitive responses. These contributions could prove indispensable in areas like mental health counselor training [6] and reddit bots [4] that are already leveraging GPT-2 in real world settings.

1.2. Data

1.2.1 Casual Conversation Data

Since our domain of interest was a subreddit that did not have any existing datasets prior, we took on the task of scraping and organizing the dataset ourselves. This was a pretty straightforward task as we used an existing data warehouse called pushshift [2] to pull information from.

We organized our data into two sets, a set of posts with unique ids and other metadata, and a set of comments that linked to posts using that unique id. This allowed us to easily organize and group the data. In particular, we scraped all data up to April 15, 2021. We scrapped a total of 522,935 posts with a total of 8,428,558 comments. We only consider

[1] https://www.reddit.com/r/CasualConversation/
[2] https://pushshift.io/

posts who have more than 10 comments and of those, we only consider the post, comment pairs at the top level. That is to say, only the top level reply is considered as a training pair. After this filter is applies, we are left with 106,646 posts with an average of 15.64 top level comments per post. This data set is feature rich and very promising for GPT-2 Finetuning. We open-source our dataset for other research endeavours here: r/CausalConversation Kaggle Dataset [3].

1.2.2 Reddit Comments

Furthering our approach to create an accurate GPT-2 model for this task, we sought out a large corpus of reddit comments that we could use to train GPT-2 from scratch over. This corpus, comprised of 1.7 billion JSON objects, was put together by pushshift and is roughly 250GB compressed. We plan to train GPT-2 from scratch using this data as well. This dataset is located here[4].

1.3. Ethical Implications

We realize that any form of intelligent and automatic text generation can have negative implications and potentially be used maliciously. We took extra care to ensure that the pipeline we develop specifically focused on positive, or neutral conversations by pulling data from and modeling r/CasualConversation posts and comments instead of from other subreddits, such as r/RoastMe.

For these reasons, we will release our dataset and architecture, but not the trained models.

1.4. Contributions

In this report we make the following contributions:

– Introduce our r/CasualConveration pipeline, CP2R, which uses a fine-tuned GPT-2 model to generate responses and post-processing to filter them, which uses a distil-bert-cased based realism discrinimator[5], and a stsb-distilroberta-base-v2 [6]) based relevance discriminator.

– A scraped dataset we realeased over r/CasualConversation, organized by posts and comments, including all reddit metadeta fields. It contains all the posts and comments up until April 15, 2021.

– An intrusion detection user study evaluating if people could detect CP2R generated responses amongst human comments based on the same post prompt.

[3]https://www.kaggle.com/rjcooper/rcasualconversation-posts-comments

[4]https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment/

[5]https://huggingface.co/distilbert-base-cased

[6]https://huggingface.co/sentence-transformers/stsb-distilroberta-base-v2

2. Approach

To create a bot that can reply to posts similar to those found in r/CasualConversation, we developed a generative and post-proccessing pipeline. In the generative aspect we fine-tuned a GPT-2 model based on data we collected from r/CasualConversation to generate comment responses based on the title of a post. Since GPT-2 responses aren't always reliable, our team also trained models to score and rank generated responses. All code for this project was made by the team itself, which utilized frameworks such as Huggingface and PyTorch. Both had examples for using pre-trained models such as GPT-2 and BERT [3], but these were changed heavily to fit our data and problem.

2.1. Text Generation

Unsupervised text generation has become increasingly powerful with the introduction of transformers, and GPT-2 serves as a deep architecture we can tune to our task. We compare three versions of GPT-2 text generation and explore their effectiveness. As a baseline, we use the existing GPT-2 model. Next, we plan to finetune GPT-2 over part of the collected data, then all of the collected data. With this finetuning, we hope to explore the effect of overfitting to our domain and show that not much finetuning is needed to get good results. We also planned to fully train a reddit-only GPT-2 model, but this proved to be too intensive with our compute capabilities and is discussed more below.

2.2. Post-processing

2.2.1 Relevancy

We can formalize the problem of relevancy as given some dataset $\{(p_0, c_0), (p_1, c_1), ...\}$ where p_i is some post, c_i is a comment of p_i, we try to see if some new c_j is relevant to some p_k. We explore two approaches to address this.

Upvote Prediction Upvotes are a method that Reddit uses to rank comments within a post. By predicting the number of upvotes, we hoped to determine relevance. Given that $\{((p_0, c_0), u_0), ((p_1, c_1), u_1)...\}$, where u_i is the number of upvotes on c_i in post p_i. We seek to use u_i as a proxy to relevance. The post and comment were both tokenized using distilbert and passed to either a pre-trained *Bert for Sequence Classification Model* or a *feed-forward regression model*. For the classification model, the upvotes were split into 4 quantiles to adjust for the significant over representation of comments with low scores.

Relevancy Discriminator Similar to the upvote prediction, we trained a model to discriminate between comments belonging to a post or not. To clarify, the training data for this task is $\{((p_i, c_i), 0), ((p_i, f_i), 1), ...\forall i \in X\}$, where p_i is some post, c_i is a comment of p_i, and f_i is also a comment, but not of p_i. Class 0 means a comment is relevant to the post while class 1 means the comment isn't relevant to the post. Notice then that the problem of relevancy becomes binary classification where for some new (p, c), and a trained model M, $M((p, c)) = 0 \implies c \in p$ and $M((p, c)) = 1 \implies c \notin p$.

Below we will describe the tasks we will train and a short description of them:

– GPT-2 for Sequence Classification using Word Embeddings: Uses the last token in order to do the classification, as other causal models (e.g. GPT-1) do.

– Bert for Next Sentence Prediction using Word Embeddings: The model receives pairs of sentences as input and learns to predict if the second sentence in the pair is the subsequent sentence in the original document.

– Feed Forward Classification using Sentence Embeddings: Using using Sentence Embeddings (stsb-distilroberta-base-v2 [7]), we generate and classify encodings. It's a three layer feed-foward network with ReLU between each linear layer and softmax at the end. Layer dimensions are $768 > 250 > 50 > 2$.

2.2.2 Realism

The realism discriminator follows a similar architecture to the models employed for our relevancy method. Similar to adversarial approaches, we first generate comments given a post as a prompt and label these as fake. Then fake comments are combined with the real comments to make the dataset for the model.

For generation, a base GPT-2 model was used to generate the fake comments. One fake comment was generated for each of 7000 post titles and a real comment was sourced from the collected dataset. Generation of the 7000 fake comments took 8 hours on Google Colab, so in order to train simultaneously, the finetuned GPT-2 model was not used to generate the negative samples, but we consider this acceptable as GPT-2 generated responses are still good enough and it gives us a baseline model that can be improved.

Our dataset consists of pairs of posts and comments. Each pair with a real comment has a class of 1, with a fake comment a class of 0. Similar to relevancy, this gives us data in the form of $\{((p_i, r_i), 1), ((p_i, k_i), 0), ...\forall i \in X\}$, where p_i is a post, r_i is a real comment, and k_i is a fake comment. This becomes a binary classification problem where for a new post and comment, (p, c), and a trained model M, then $M((p, c)) = 1 \implies c \in p$ and $M((p, c)) = 0 \implies c \notin p$. We used Bert for Sequence Classification on distil-bert-cased[8], this was trained to discriminate between fake and real comments to provide a measure of realism for newly seen post-comment pairs.

2.3. Pipeline

Following our original idea of creating a bot capable of responding to r/CasualConversation posts, we've outlined the different approaches we used to solve this problem. Based on our experiments for response generation and post-processing our final pipeline, CP2R, can be shown in Figure 1. Every generated response is cleaned and checked for validity, then scored based on its realism and relevancy to the prompt. We keep generating possible responses until we have x responses above a realism threshold and relevancy threshold, both set to 0.9 (out of 1). Out of the x responses above our thresholds, we pick and output the best response by finding the one with the highest product of realism and relevancy scores.

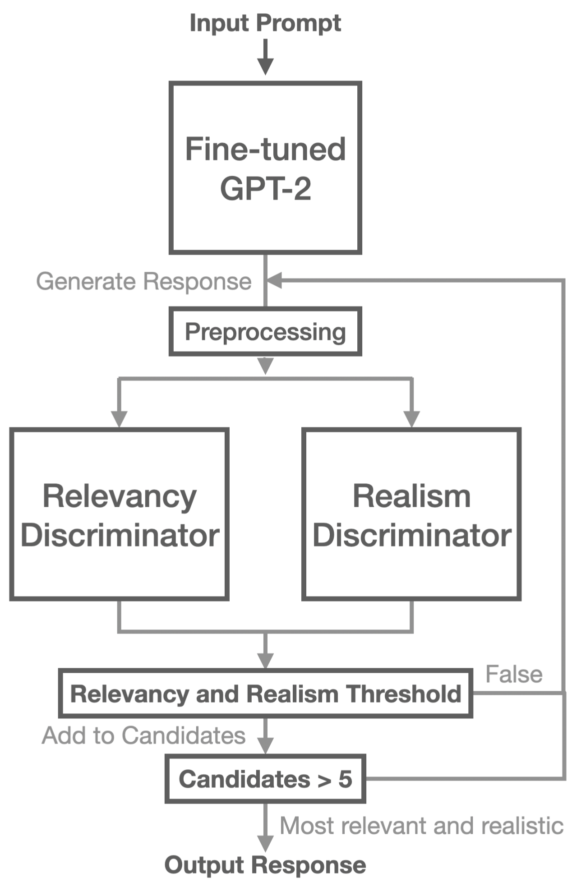

Figure 1. Our entire CP2R (CausalConversationBot) pipeline.

2.4. Problems

Reddit data consists of diverse elements including emojis, links, pictures. Furthermore, the Reddit language is doesn't follow the traditional grammar rules that GPT usu-

[7]https://huggingface.co/sentence-transformers/stsb-distilroberta-base-v2

[8]https://huggingface.co/distilbert-base-cased

ally outputs. To combat this, we attempted to use a realism discriminator and a relevancy discriminator. Realism performed well, but our initial relevancy models weren't able to identify relevant comments. After changes in architecture, training tasks, and hyperparameters we were able to develop a model capable of picking out responses relevant to given prompts.

We anticipated compute power as a big issue for the pace of the project. As an example, even with the P100 GPUs provided by Google Colab, GPT-2 still took 30 hours to finetune and each of the discriminators about 6 hours each. These models were also trained multiple times using different hyperparamerters and this adds to the computational demands of this project. To address this from the start, we compartmentalized as many tasks as possible so that each can be trained concurrently. For example, the realism discriminator was trained on samples generated from the base GPT-2 instead of our fine tuned model.

Another major issue we encountered was that we were not able to train GPT-2 from scratch in a realistic amount of time. We sourced 250GB of reddit data specifically for this purpose, but had to cancel the training after a week since it was not going to be done by the deadline of this project.

3. Experiments and Results

In this section we'll go over the performance for our models and an intruder study evaluating our finished r/CasualConversation bot pipeline, CP2R. To evaluate model performance, we report the validation loss and testing scores of our models and include discussion about why different implementations may have failed or succeeded.

3.0.1 Training Hardware

For the majority of our tasks, we utilized Google Colab Pro's P100 GPUs. For training GPT-2, we utilized a DGX-2 compute cluster available to our team.

3.1. Text Generation

After finetuning the GPT-2 model on our entire corpus, we arrive at a training perplexity of 2.1970, which is considerably low for the task at hand.

In the rest of this section we will explore differences between the fine-tuned model, a partially fine-tuned model, and stock GPT-2 for text generation. Consider a standard conversation starter, "How was your day?". We will generate candidate responses with each language model and present the sample as a representation.

From Figure 2, we see that as we increase the finetuning on GPT-2, the results begin to make more sense. Note, these responses do not have any processing done to them and they are just five random samples. We identify that increase in performance is due to the way we finetune GPT-2,

GPT-2 (Stock)	GPT-2 Finetuned (20%)	GPT-2 Finetuned (Full)
I know it wasn't too awesome, but at least it didn't mean you couldn't have dinner with some jerk, or some kid or some asshole.	Aw man I feel I need to go to sleep, how ya doin? Woke up to something really good, slept into work like usual though. Feeling pretty damn good.	It's 7:14PM here and the day has been amazing in spain so far. Had to stay up all night looking for the office to get up at 2:21 a.m..
I was just tired, so I didn't do anything at all. I woke up late, and then had another 20 minutes to get up and do my work. It felt very sick at the time.	Been chilling out, listening to some podcasts on my phone. I have two more weeks, and I just finished Breaking Bad.	Just drinking a can of Budweiser at home and browsing some reddit when the day was finally over. Good night everyone.
Yeah . I woke up a bit late. (laughs) I'd been looking at it one morning and I had been browsing through the posts.	Well, yesterday I went to lunch with her, she gave me a banana on my left side and I walked to my next shift, she gave me a pizza for lunch, and I ate lunch.	Woke up at 3PM and went to school. No class for another 24 hours.
I'm kind of stressed out on Monday nights but I've been doing something productive and I'm getting a lot of sleep lately	I had a really nice day! So what were you doing/thinking on this lovely day this time of year? I am so happy for you all! :) :)	I'm still trying to comprehend why I ate so much spaghetti!
I'm going out tomorrow. I'm going to eat dinner today.	I didn't sleep last night, but my girlfriend's birthday is next week and I decided to celebrate it with her.	Day was alright. Went to a wedding.

Figure 2. Comparison sample of Stock GPT-2, Partially Finetuned GPT-2, and Fully Finetuned GPT-2 on a sample prompt, "How was your day?".

moving the problem from next token generation to sequence inference helps improve generation. In Figure 3 we see the results of the same prompt using our CP2R pipeline.

CP2R
Tired. Working. But it's going well.
I'm a night owl and my boss lets me sleep in between 9:30 and 3:00 (11 am on a Wednesday).
Work. So exhausted at home. Trying to decide what to do with my life.
My day is finally over. I'm just sitting here on my computer and browsing reddit!
Pretty chill! Today has been an exciting and productive day.

Figure 3. CP2R random sample from sample prompt, "How was your day?". The italicized response was chosen as the pipelines output. Notice that CP2R's responses are quite conversational and realistic while being relevant to the prompt.

3.2. Post-Processing

Even after fine tuning, the GPT-2 model still outputs nonsensical results at times. The grammatical structure of the sentence was usually correct, but some responses were generally lacking in relevance to the prompt (post title) or could easily be identified as machine generated.

3.2.1 Relevancy

For upvote prediction, neither GPT-2 for Sequence Classification nor Bert for Next Sentence Prediction tasks converged so these methods were not used in the final pipeline. Both ended with a loss of around 0.71 and low accuracy.

We speculate that this could be because upvotes do not necessarily dictate relevance. Consider that a low ranked post will have low ranking comments despite them being relevant. Normalizing the comment score with respect to the post score did not result in better prediction. One thing to note is that both BERT-based upvote prediction models outputted the same values regardless of the input.

We tried a few iterations of the relevancy discriminator. Using both GPT-2 For Sequence Classification and Bert for Next Sentence Prediction failed to converge to any reasonable loss. We credit this to the complexities in these task and the amount of data needed to get these tasks to converge. With the final model, Feed Forward Classification using Sentence Embeddings, we attain a final f-1 score of 0.75880, which given the complexity of the problem space, we consider sufficient. We believe that this model is severely under-trained, and consider the addition of more data to be future work. This model is the one used for Relevancy discrimination in CP2R.

3.2.2 Realism Discriminator

Using a sample of 13,315 mixed true comments and generated comments, the model converged with a final loss of 0.1492 starting from a loss of 0.7. This is considerable, and shows that the data is quite separable for a binary classification task. We consider the model to be severely underfit, and in need of more training samples from GPT-2, but these are computationally expensive and time consuming to generate.

3.3. Hyper Parameters

- Finetuning GPT2: 2 epochs, 500 warmup steps, 400 eval steps

- Realism and Relavancy: default BertForSequenceClassification parameters

- Feed Forward: CrossEntropyLoss, and learning rate of 0.0001

3.4. GPT-2 Casual Conversation Intruder Study

To determine how realistic the outputs are, a user study was conducted using a multiple choice questionnaire. Recent posts were selected from the subreddit r/CasualConversation that were not apart of our training data, along with the four top comments in the thread. The post's title is passed through the pipeline to get CP2R's top generated responses. Within the CP2R pipeline, we consider 5 candidate responses and choose the response that maximizes the product of the realism and relevance scores. The combined five responses, four human generated and one machine generated, are put as multiple choice answers for each post in the form. We used this method on 10 total posts for the user study. Users were asked to choose the option that is most likely to be AI-generated.

Task	Conversation Query
0	How was your day, Reddit?
1	After a year without Instagram or Facebook, I'm shocked that I ever shared personal details of my day with hundreds of people.
2	Is it bad to visit our parents way to often?
3	As of this morning I am a published author!
4	I asked for a raise and I got it.
5	I just had a fresh mango for the first time in my life.
6	I showed up late for work yesterday and everyone was genuinely happy to see me.
7	Ate a banana with the peel.
8	An old friend committed suicide yesterday.
9	Is anyone else going to continue to wear a mask after the mandate is lifted?

Prompts for each of the tasks in the User Study.

155 responses were collected using a Google form[9]. Participants came from diverse backgrounds concerning age, technical proficiency, and Reddit exposure.

Of the 10 task, the median identification score was 4. the modal score was 2, and 9 of the 10 questions had a plurality of people correctly choosing the machine generate response. However, only 2 of the 10 questions achieved a majority of of people choosing the machine generated response. Fascinatingly, the plurality of users selected the machine generated response for the vast majority of questions which seems to suggest that their are some identifiable features of these generated texts that are being detected in aggregate, but not individually. It is interesting to note that no single participant got every task correct.

Task	Average Rate of Detection
0	0.3768
1	0.4420
2	0.3043
3	0.6014
4	0.2029
5	0.3406
6	0.3913
7	0.4855
8	0.4420
9	0.4203
AVG	0.4007

Table 1. Task-specific results of Intruder Study

We also compute the Krippendorff Alpha statistical measure of the agreement of multiple reviewers over multiple

[9]https://forms.gle/MiSyc8iYYHmAF8B59

tasks. We find our data shows that $\alpha = 0.03701$, which is indicative that there is an absence of reliability between reviews. This is a promising result as it implies that there is little agreement among individual respondents on consistently identifying the machine generated results. Combining this with our observation that aggregated participants, per the wisdom of crowds, seem able to detect the machine responses with relatively high accuracy (90%), it seems that individually people have biases about what machine generated text should look like, but in aggregate these biases appear to balance out despite the fact that only two of our participants scored 9 out of 10 on the study individually.

No formal qualitative study was conducted, however lots of respondents were enthusiastic and wanted to discuss their thoughts. We think this enthusiasm enabled the survey to go viral and contributed to the success of this study. Many were impressed and wanted to compete with their friends. Common responses included the following identification strategies:

– If not told there was a fake response, they would not have questioned any of the responses if seen on the Reddit website

– Some made assumptions of emoji/special character capabilities and ruled out some responses

– Grammar is a give away

– Machine generated responses tend to be less specific to the prompt

3.5. Drawbacks

Our initial goal was to construct a text-based response generation agent that was biased to generating kind or friendly responses. We believe we were partially successful in this goal with most of the responses we generated being be free of any animosity or hostility. The only problem that we encountered was that some of the responses to serious prompts would be inappropriately positive. For example, to the prompt "An old friend committed suicide yesterday" one of the selected responses was "I'm sorry to hear that, OP :)." Clearly, this response could be considered 'nice' at a semantic level, but at a pragmatic level it clearly falls short of offering a truly supportive response as the smiley face could be considered inappropriate for such a somber post.

4. Conclusion

Our pipeline successfully demonstrated that realistic responses can be generated using the underlying GPT-2 architecture and several post-processing models. Future work could more robustly optimize our three models and better filter for kind comments that are inappropriate for a specific context. Despite these limitations, we see that our model is able to trick real users with the median user from our user study only detecting 4/10 machine generated responses and the modal user only detecting 2/10 machine generated responses. We find that the model was only partially successful in blending in with real comments with the wisdom of crowds easily selecting for the machine generated comments, but individually users noted that if they were not made aware of the presence of machine generated text they likely wouldn't have suspected anything. Finally, we believe that we were only partially successful in constructing a "kind" text generation agent with the model at times responding with inappropriately peppy responses to serious prompts. Turns out kindness is contextual.

4.1. Code Repository

Like was mentioned earlier, none of the trained models will be released, but the code for model generation and training can be found at: https://github.com/Coopss/cs7643_project.

Student Name	Contributed Aspects	Details
Ryan Cooper	Data Collection GPT-2 Finetuning Realvence Discriminator Pipeline	Wrote code to scrape data for Reddit using pushshift. Designed the realvence discriminator, implemented in PyTorch. Trained GPT-2 Finetuned model on reddit data. Finished the pipeline implementation started by Jonathan.
Abhishek Mallemadugula	Background Research Realism Discriminator Study Design	Researched existing techniques in sequence and text generation. Designed, trained, and implemented the realism discriminator. Designed user study and found questions to ask.
Christain Bolyston	Data Collection GPT-2 Finetuning Relavence Discriminator	Gathered large corpus of reddit data (250gb) for training GPT-2. Trained GPT-2 Finetuned model, experimented with configurations. Implemented relavence model in sklearn
Jonathan Leo	Background Research Upvote Predictictor Pipeline	Background research for GPT-2 discrimination techniques. Wrote code for initial upvote predictor, relevancy model, and pipeline. Analyzed results of upvote predictor, determined it may be too difficult.

Table 2. Breakdown of Contributions by each team member.

References

[1] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[2] Paweł Budzianowski and Ivan Vulić. Hello, it's gpt-2–how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. *arXiv preprint arXiv:1907.05774*, 2019. 1

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[4] Will Douglas Heaven. A gpt-3 bot posted comments on reddit for a week and no one noticed, Oct 2020. 1

[5] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*, 2020. 1

[6] Abby Ohlheiser. An ai is training counselors to deal with teens in crisis, Mar 2021. 1

[7] Bo Pang, Erik Nijkamp, Wenjuan Han, Linqi Zhou, Yixian Liu, and Kewei Tu. Towards holistic and automatic evaluation of open-domain dialogue generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3619–3629, 2020. 1

[8] Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. Automatically neutralizing subjective bias in text. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pages 480–489, 2020. 1

[9] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1

[10] 2020 — TechForge Media Categories: Healthcare Ryan Daws — October 28, Adoption, Bots, Gaming, Applications, and Ethics. Medical chatbot using openai's gpt-3 told a fake patient to kill themselves, Oct 2020. 1