Conversations on Discord: Integrating Chatbots with the Discord API and its Ethical Implications

By Teera Tesharojanasup and Matthew Goldgirsh

Abstract

Chatbots are becoming vastly more popular with the creation of OpenAI's ChatGPT. With its newfound popularity, the development of chatbots sparks questions of what functionality chatbots should have, and what they should be used for. This paper investigates a chatbot that operates over the Discord platform, a communications application usually used for entertainment purposes. The chatbot will first be constructed using two different architectures and two different datasets for variability. Using these varying architectures, responses are generated and the coherence is investigated as well as the ability to maintain conversation. This data will be analyzed and there will be a commentary on the importance of "good", ethically reasonable, data more so than the architecture chosen.

1. Introduction

The project is based on developing a chatbot integrated with Discord with the ability to fluidly answer users' questions as well as have conversations when prompted. The success metrics will evaluate the response time of the chatbot and its comprehensibility. For the purposes of the project, it was planned to only train one model based on the daily dialogue dataset available on Hugging Face. This model was named DiscordGPT. Once the chatbot is trained, it is integrated with the Discord API so users from the popular app can interact with it.

However, halfway into the project, we wondered about the ethical implications. If software developers are able to train chatbots to maintain a coherent conversation, could they also train them to generate misinformation, divisive political commentary, and hate speech? With limited resources, the project's Discord chatbot will use a small labeled dataset based on 4Chan's politically incorrect page to train a second model named DiscordChan.

2. Methodology

In order to analyze the implications of chatbots, extensive research was needed to find what datasets to use and what model architectures to implement. Furthermore, since the project will investigate the ethical implications of chatbots, it is important to find datasets that demonstrate the good and bad dialogues that a chatbot may have. The research conducted placed a strong emphasis on the discrepancy between the "good" and "bad" datasets rather than the model architecture to evaluate the hypothesis and the ethical implications of datasets.

For the "good" dataset the model, DiscordGPT, was trained over 3 epochs with a learning rate of 5×10^{-5} with a cosine annealing scheduler. The model also was trained with a 0 weight decay and a batch size of 4 per device. Furthermore, it was trained using a subset of the "good" dialog dataset where there are only two actors in the conversation. This subset was used due to the nature of conversations on the messaging platform Discord since they only involve 2 actors: the user and the chatbot.

The training procedure for DiscordGPT looked like the following. Given a list of sentences with two actors (speakers), first, the sentences would have to be cleaned into the form shown below.

```
"<|USER|> Hello, the weather is nice today. <|COMPUTER|> Yes, it is”
```

Then a pretrained openai/gpt2 byte pair encoding tokenizer would be run on the sentence to produce a vector of features. These encoded byte pair vectors were then funneled into the model with the training arguments specified. Once trained, to generate values the encoded context of the sentence would be put into the model and the response encoding would be generated which would then be decoded by the pre-trained tokenizer and shown to the user using the Discord API.

In regards to the model trained on the 4Chan dataset termed DiscordChan, the training data was formatted into tag-pattern-responses format where the model learns a pattern to the user's input. Due to the small dataset, it was decided to not split up the data any further into test and validation sets.

```
{
  "tag": "greeting",
  "patterns": ["Hi", "Hello"],
  "responses": [
    "Hello, I hope you're ready to see the worst chatbot in existence",
    "Bye",
    "Stop, don't talk to me."
  ],
  "context": [""]
},
{
  "tag": "random",
  "patterns": ["Give me a fun fact", "Another one", "I want to hear more"],
  "responses": [
    "The First Computer Programmer Was a woman.",
    "The Japanese army in WWII was evil",
    "Swearing isn't swearing if you use it enough"
  ],
  "context": [""]
}
```

In the example above, the model learns that the word “Hi” and “Hello” is the greeting tag. Given the user input “hii” the model will choose the “tag” with the highest probability. Once chosen, the model randomly selects one of the three responses prepared under the greeting tag.

To learn how to give an appropriate response, DiscordChan uses a Sequential Neural Network where the layers are stacked sequentially on top of each other. There are three dense layers consisting of 128, 64, and 6 neurons respectively. The first and second layer uses the ReLU activation function which returns 0 for any negative input and returns 1 for any non-negative input. The third and final layer uses the softmax activation function which returns a probability distribution. In between the dense layers, dropout layers are added to prevent overfitting which randomly sets a fraction of the input units to zero during training. Finally, the model is compiled using Categorical Cross-Entropy or CatCE where CatCE measures the difference between the true distribution of the data versus the predicted distribution given by the model. The model is trained for 200 epochs with a batch size of 5 and saved.

3. Data

The “good” dataset, used to train DiscordGPT, was taken from Hugging Face. This dataset called “daily_dialog” is very commonly used in the field of natural language processing to train generative models. The daily dialog is commonly used because it is human-written and reflects daily conversations talking about daily life. The Dialog dataset consists of annotations on the emotions expressed per utterance in the dataset as well as annotated with what the actor said in the utterance in the conversation.

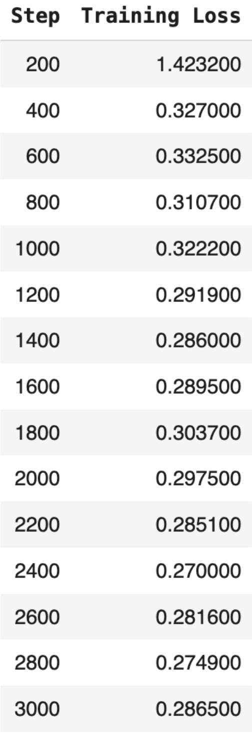
For the model, this dataset was filtered based on the number of actors partaking in the conversation. Since the model should be able to maintain a conversation between itself and one other individual, the training data was filtered to where there were only two actors present in the conversation. From there the sentence was labeled with two distinct tokens, the user token and the computer token. The user token was used to prefix the sentence that the first character uttered and then the computer token was used when the actor changed to prefix the next sentence denoting what was said by the computer.

To train DiscordChan, the second model required labeled data consisting of phrases and sentences paired with corresponding tags. While various datasets with labeled data were accessible, none of them included information sourced from 4Chan. This absence is logical considering the potentially malicious applications that could arise from utilizing data from 4Chan.

To give context, 4Chan is an anonymous online forum renowned for its diverse user base and unrestricted content. Since its inception in 2003, the platform has hosted discussions on a broad spectrum of topics, including video games, anime, politics, and conspiracy theories. Notably, certain boards, such as /pol/ (Politically Incorrect), have gained infamy for harboring extremist views and discussions, including white nationalism and hate speech. The anonymity provided by 4Chan facilitates the proliferation of such ideologies, as users can engage in controversial discussions without fear of accountability. While not all users or boards promote extremism, the platform's association with such content raises concerns about its impact on online discourse and society. 4Chan essentially tries to navigate the fine line between preserving freedom of expression and preventing the spread of harmful ideologies.

For the purpose of this paper, data sourced from the dataset *Raiders of the Lost Kek: 3.5 Years of Augmented 4Chan Posts from the Politically Incorrect Board* required manual labeling. The process involved selecting sentences and assigning them to various tags, creating a small dataset suitable for demonstration purposes only. Expanding the labeled dataset any more would enhance the capabilities of a potentially malicious chatbot, blurring the lines of legality and is outside the scope of this project.

4. Results



Step	Training Loss
200	1.423200
400	0.327000
600	0.332500
800	0.310700
1000	0.322200
1200	0.291900
1400	0.286000
1600	0.289500
1800	0.303700
2000	0.297500
2200	0.285100
2400	0.270000
2600	0.281600
2800	0.274900
3000	0.286500

DiscordGPT's training loss as a function of step count is shown above. It is evident from the training loss that the model is getting trained as the epochs continue and the training loss is going down.

In order to analyze the model further the Bleu score was calculated based on a sample of 10 sentences. The bleu score was calculated to be 0.005567 which is a very poor score for a generative model. The very low Bleu score calculation however can be accounted for because only a sample of 10 sentences was used in the calculation and hence the dataset was not large enough to provide an accurate score. Furthermore, the Bleu score is not the best metric to measure generative chatbots as it measures the exact similarity between references and generative wording rather than the semantic similarity.

To analyze DiscordChan, a few metrics were calculated on the small dataset.

Class	**Precision**	**Recall**	**F-1 Score**	**Support**
0	1	1	1	5
1	1	1	1	2
2	1	1	1	5
3	1	1	1	2
4	1	1	1	3
5	1	1	1	5
Accuracy			1	22
Macro Avg	1	1	1	22
Weighted Avg	1	1	1	22

Due to the small dataset used to train DiscordChan and the absence of the testing and validation sets, the following classification report should be taken with a grain of salt.

From this results tab, it is important to note that due to the nature of generative chatbots, it is difficult to generate good evaluation metrics that demonstrate the effectiveness of the chatbot; rather it is more important to see a demonstration of the chatbot to see its cohesion and clarity.

5. Discussion

The development of the DiscordChan chatbot has shed light on both the capabilities and limitations of current chatbot technology, particularly when trained on datasets with questionable content. As evidenced by the provided architecture and training data, the chatbot's functionality is currently constrained by several factors such as the need for manual labeling, and hardware limitations. However, it is crucial to recognize that these constraints can be lifted with access to greater resources and using unsupervised learning methods, raising significant ethical concerns regarding the potential misuse of advanced chatbot technology.

The DiscordChan chatbot is trained on a small labeled dataset sourced from the politically charged environment of 4Chan's /pol/ board. While this dataset provides a glimpse into the potential capabilities of the chatbot, its size and quality are greatly limited. The manual labeling process, necessitated by the absence of readily available labeled data, introduces subjectivity and potential biases into the training process. Moreover, the dataset's small size restricts the chatbot's ability to generalize and respond to a wide range of user inputs.

The training process of DiscordChan relies on hardware resources that may be suboptimal for handling large-scale datasets and complex neural network architectures. With limited computational power, the chatbot's training time and model complexity are constrained, resulting in a less sophisticated and capable final product. As a result, the chatbot will struggle to generate coherent and contextually appropriate responses, particularly in nuanced or ambiguous conversational scenarios.

DiscordGPT faces similar challenges, particularly in how datasets often mirror the biases and linguistic nuances of their creators. This phenomenon significantly impacts the ethical dimensions surrounding the chatbot. For instance, a comparative examination of datasets like 4Chan and daily dialog demonstrates the influence of distinct datasets on shaping both the responses and the general tone of the chatbot.

While the existing constraints of DiscordChan and DiscordGPT alleviate immediate concerns regarding misinformation and harmful content, the scalable nature of chatbot technology raises profound ethical dilemmas. The potential for DiscordChan's expanded capabilities extends far beyond the confines of Discord, offering a tool for malicious activities such as manipulating political discourse, disseminating misinformation, and subtly promoting propaganda. As chatbots increasingly resemble human speech, the necessity for accountability, regulation, and ethical oversight becomes imperative.

Further exploration into the ethical implications of chatbots necessitates training models with diverse datasets and architectures to uncover prevalent biases. Research should be done into the effectiveness of various tokenization, lemmatization, and semantic analysis techniques in mitigating biases and facilitating unbiased information dissemination, particularly in LLM like ChatGPT.

6 References

Li, Yanran, et al. “DailyDialog: A Manually Labeled Multi-Turn Dialogue Dataset.” *Datasets at Hugging Face*, Hugging Face, 2017, huggingface.co/datasets/daily_dialog.

Chudoba, Michal. “Daily Dialog GPT + Dialog System.” *GitHub*, Jan. 2024, github.com/jinymusim/Daily-Dialog-GPT.

Wang, Ben, et al. “Mesh Transformer JAX.” *GitHub*, 2021, github.com/kingoflolz/mesh-transformer-jax/.

Kilcher, Yannic. “GPT-4chan: This Is the Worst Ai Ever.” *YouTube*, 3 June 2022, www.youtube.com/watch?v=efPrtcLdcdM.

Griffo, Umberto. “Minimalistic Multiple Layer Neural Network from Scratch in Python.” *GitHub*, 2017, github.com/umbertogriffo/Minimalistic-Multiple-Layer-Neural-Network-from-Scratch-in-Python/tree/master.

Papasavva, Antonis, et al. “Dataset: Raiders of the Lost Kek: 3.5 Years of Augmented 4chan Posts from the Politically Incorrect Board.” Zenodo, 13 Jan. 2020, zenodo.org/records/3606810.

7. Appendix

> Structure of the code and user manual refer to the readme.md on https://github.com/mgoldgirsh/nlp-discord-bot.

> External dependencies and libraries are in requirements.txt

> Description of functions/methods/classes in code comments.