

# Final training results

## MAIS 202 Deliverable 3

**Sarvasv Arora, Kaiwen Xu**

March 28, 2021

### The model and some results

We finetuned the GPT2 pre-trained model provided in the Hugging Face library.

Due to Colab GPU limits, we couldn't use the entirety of the data for the same, however, the outputs produced are still decent enough as inspected by us on several random comments.

We also tried different types of text generation methods like beam search, top-p sampling, greedy search, etc. and found that beam-search generated the best replies as we perceived by testing it on a bunch of randomly selected comments.

Top-p sampling also performed well, but we liked the replies of beam search more.

Following are some parent-reply pairs as produced by beam search sampling method:

- Comment: You are on the internet, you can watch everything somewhere.  
Reply: That's the only reason you're not going to have a good life, because we all know that.
- Comment: Everyone glared at me like we were together and I was now responsible for her outburst. Thanks, lady.  
Reply: Because it's a woman's fault, right?
- Comment: Muslims should go back to the middle east where Jesus came from!  
Reply: Yeah, but it's not like the middle east is a country.
- Comment: There are two tricks in math: adding zero and multiplying by one. [sic. Prof. Tim Hoheisel, MATH 247]  
Reply: That's why you can't be a math teacher, because you don't understand math, you're just a mathematician, and you have no idea of how math works, so you must be wrong.

As observed, the replies are decent, given that we could only finetune the model on a very small set of replies. With more computing power and a better heuristic of categorizing sarcasm, the model can be improved a lot!

### Final demonstration proposal

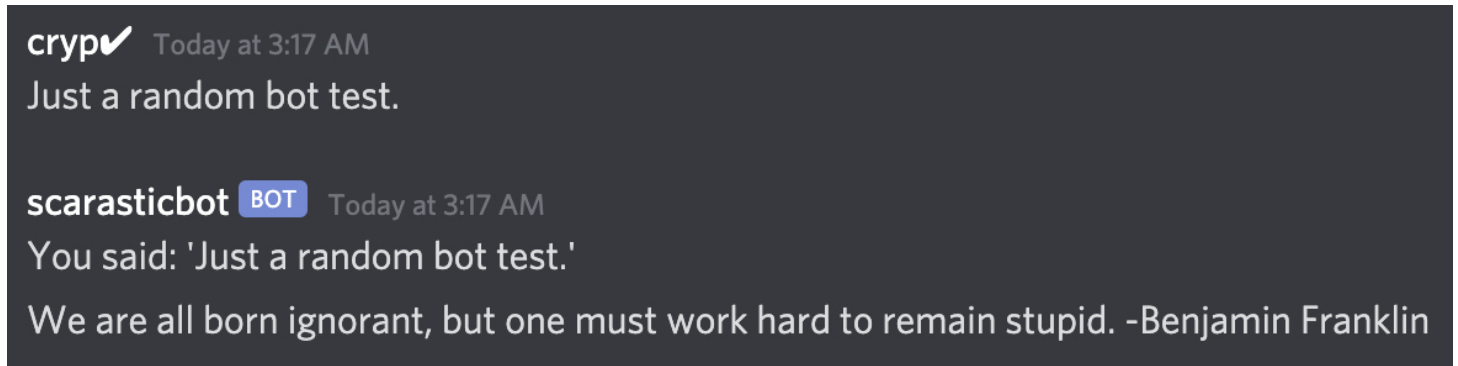
Nowadays, the instant messaging app Discord is extremely popular among young people. Imagine that you are chatting with your friends on Discord, and suddenly a cute robot joins the conversation with a sarcastic, but very amusing comment on your words. That will certainly make you laugh.

So, we decide to create a sarcastic comment Discord chatbot as our final product. Using Discord APIs, we can read messages from users. Then we will feed the user's message into our model, generate a related sarcastic comment, and send the comment back to Discord channel. We will deploy our code on our own remote server.

Data privacy is respected since we don't store the users' messages. The messages remain in our server as long as the model makes an inference (i.e., generates a reply) and are unloaded from the memory as soon

as the script dies.

The following photo gives an example of a testing bot that output an inspiring quote whenever it receives a message. We will integrate our predicting model with the bot once our model is ready for deployment.



Another feature that we plan to add is producing replies to specified comments. So, a user would be able to tag the bot as a reply to the comment, and in response the bot will only produce a reply to that comment.