Capstone Proposal - Chat Moderator

Running online communities can be really challenging. Those in charge generally want a place where people can come and feel welcomed. Unfortunately, the internet is filled with people who enjoy doing whatever they can to hurt that sense of community either out of spite or genuine hatred. While some behavior is easily detected, such as swearing and racial slurs, others are much more difficult to detect such as bullying and hate speech. The goal of this project is to use known datasets of negative speech to train a model to help detect comments that can be flagged in close to real-time for online chat communities. The power of this project is that as it gets more sophisticated it can allow easier creation of all kinds of online chat communities where members can feel safe to go and not worry about the negativity of those less interested in building a positive atmosphere.

The challenge and interesting parts of this problem come into play in both the deep learning as well as the implementation of the chat system itself. The first hurdle is actually training the model that can detect and register chats as being disruptive in a channel. The second hurdle is around scaling the system to be able to quickly run the detection of the chat through it such that it can respond in meaningful time. Finally, the last hurdle is the integration with the chosen chat system itself. The most common for gaming communities in recent years is Discord so this will be avenue taken for the project.

The cleanest dataset that seems recent enough to work with this problem is from Kaggle based on the paper "Automated Hate Speech Detection and the Problem of Offensive Language." by Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber (2017). The dataset found here https://www.kaggle.com/eldrich/hate-speech-offensive-tweets-by-davidson-et-al is well labeled and has multi-class labels for both offensive and hate speech giving the option of more fine-grained control over the level of tuning the bot can have.

In order to solve this problem, the data will be processed and explored with Pandas. From there a deep dive will be taken into the Spacy library which is a strong standard package to attack NLP problems allowing for many standard techniques for breaking the text up into features that can be used for the actual machine learning. The first pass at classification will be a logistic regression with the features chosen through Spacy. The hope is to get at least 70-80% accuracy with the dataset that has been chosen. Once tuned the next step is moving into more deep neural networks to see if higher accuracy can be gained without the loss of too much speed in predictions.

Using these approaches, the system should be able to somewhat accurately predict if any given statement is offensive or hate speech on a deeper level than simple heuristics. It can also find more terms than just ones in a list of offensive words that may change over time.

At the end of the capstone project, the goal will be able to run a successful test of a chatbot using the trained model in real-time. Once the chatbot is added to the Discord community it will monitor requested channels scanning each message for offensive language then warning the user when they make the offensive comment to please refrain. After a series of warnings, the user will be suspended from the chat for a period of time before being banned from the server. Besides the chatbot itself, the model will need to be hosted in a place that can run it rapidly for the chatbot to monitor large amounts of traffic. The prediction time needs to remain under 5 seconds in order to keep the reprimands timely and not get lost in the conversation. There is a high likelihood that this will also involve an API to configure the bot as more of a stretch goal.

The two main resources for the final product will be the compute to train the model then the lesser compute to make classifications against the model to determine the class of speech. The entire dataset in its raw form is only 7 mb which will expand as various methods are employed to make the features more viable for machine learning. Estimating the expansion there shouldn't be more than about 1 gb of ram needed at any given time in training or predicting on the model. As for training the model in the deep learning method, it shouldn't take more than 8 GPU hours to fully train the model with the training set available.