

# Proposal: To Create a Chatbot for Detecting Hate Speech

Wesley Aldridge  
waldridge@knights.ucf.edu

Miles C. Crowe  
miles.crowe@knights.ucf.edu

Mauricio De Abreu  
mabreu@knights.ucf.edu

## 1. MOTIVATION & PROBLEM STATEMENT

Hate speech is defined broadly as aggressive language targeted at a person for attributes typically beyond their control. Traits such as nationality, gender, race, sexual preference or disability are popular targets of hate speech[1]. Aggression towards the victim may manifest as insults, personal attacks or even threats. As online communication between people is becoming more commonplace, exposure to hate speech is almost certain to reach more people. Unfortunately as it is, this allows the spread of underlying beliefs that contribute to such behavior.

The motivation for this project is rooted in the desire to curtail the spread of hateful language. There are countless paths for hate speech to spread. Providing a human intervention is infeasible given the amount of communication that occurs over the internet. Naturally, this presents an excellent opportunity to contribute an automated approach to detecting and flagging hate speech by utilizing NLP in real time.

## 2. RELATED WORK

Hate speech has earned enough interest to inspire governments into getting involved[2] by passing laws explicitly prohibiting hate speech. Given this, many have examined various methods for hate speech detection.

Manoel Horta Ribeiro, et al.[3][4] used a Twitter hate speech dataset to characterize Twitter users as either hateful or not, in order to analyze patterns and trends among the users labeled as hateful. Instead we will use this dataset, among others, to train a neural network to detect hate speech, and we will use this neural network to create a Discord real-time hate speech detection bot.

Davidson et al. went further into separating offensive speech from hate speech. Their approach used hate speech compiled by Hatebase.org, consisting of a lexicon of both words and phrases, identified by internet users. Following their lead, we intend to use Hatebase.org's academia API to obtain modern hate speech examples. This is important as

a major shortcoming noted by MacAveney et al.[5] is those who spread hate speech are well aware that their views and texts are being suppressed and removed, and continue to evolve to evade detection.

## 3. STOPPING THE SPREAD

This project aims to curtail the spread of such speech by detecting hateful dialog and providing an alert mechanism to raise events for handling the occurrence automatically as it happens. Clearly stated, the chat bot would be a silent participant in a chat, regardless of the number of participants, which would monitor the text exchanges. When hate speech is detected, the event would be passed to a host framework or external agent to handle the event.

A simple implementation of this would be a chat bot, connected to a Discord server which is trained utilizing a robust hate speech data set. A neural network would be trained to detect hate speech. This neural network would be the critical part of the chat bot which would read user messages and detect hate speech in real time, and alert the moderators and admins of the Discord server.

## 4. EXPECTED OUTCOMES AND RISK MANAGEMENT

As in most software projects, expectations will need to be curtailed and priorities need to be established. At the bare minimum, this project should yield a simple input/output text classifier that is accurate with meeting the expectation of identifying hate speech. Formally stated, this implies that core functionality should be established prior to moving towards the novel aspects such as portability and integration with external platforms. This can be realized through a simple design that exposes only the necessary API to validate and test the outcome of the core goal. Once this goal is achieved, a narrow set of integration targets can be established for demonstration purposes.

In terms of the main goal, there is risk of biases and over fitting of the training data. Over-fitting errors will need to be carefully analyzed and bias will need to be mitigated with tools we will discover later in this course.

Another risk would be training data mismatch with regards to the platform. For example, users in Discord are allowed 2000 characters vs 280 characters for a Tweet on Twitter. Size constraints may dramatically affect how hate speech is formulated due to the compression of ideas on platforms that severely restrict text lengths. Careful validation can easily address this by restricting the test data to mimic

similar sizes to that of the training data.

Lastly, temporal changes in hate speech may present classification errors as slang and dialect may change over time. This risk will have to be accepted unless fresh and up to date training data becomes immediately available.

## 5. PLAN AND ROLES OF COLLABORATORS

Wesley will code the neural network, train it with the hate speech data, and test it. The neural network will be what the bot uses to label messages as hate speech or not hate speech.

Miles will integrate the neural network into an online hate speech detection service that he will code and test. The service will be used by the Discord bot to do its message analysis and hate speech detection.

Mauricio will incorporate the online hate speech detection service into a Discord server bot that he will code and test. The bot will detect hate speech in Discord server messages in real time.

Everyone will work on the write-up and presentation.

For the first three weeks, beginning upon approval of this proposal, the main focus will be on gathering training data, designing and coding the framework and resolving architectural challenges. The next four weeks will consist of building the neural net, training, testing and optimization. The final three weeks, or remaining time to completion will focus on integration with chat bot client, analysis of hate speech detection and writing of the report and presentation.

## 6. PROGRESS AS OF 3/3/2020

Wesley has accomplished preliminary work on building an initial neural net and some training. Results from the neural net have been inconclusive as of yet. This is due to the classification of the training data. The training data is not classified into HATE/NOT HATE, rather is it scaled from normal speech through various levels of offensive language to various levels of hate speech.

Miles has downloaded the entire data set of hate speech vocabulary from hatebase.org. This vocabulary can be used to annotate speech as containing hate speech indicators. Miles also created a Python based Discord bot that is a member of our group discord. Design of the API therein to mesh the Python bot to Wesley's classification system is underway.

Mauricio has created a NodeJS implementation of a Discord bot in parallel to the work that Miles has accomplished. Mauricio was able to complete some rudimentary behaviors such as warning, kicking, etc. This affords the project flexibility if we want to switch to a NodeJS implementation.

Challenges moving forward will always be time availability and lack of training data. Each member will continue to gather more examples of Hate Speech so that a more robust classifier will ultimately be created.

## 7. REFERENCES

- [1] "Dictionary.com." <https://www.dictionary.com/browse/hate-speech>. Accessed: 2020-01-21.
- [2] M. Davidson, Warmesley and Weber, "Automated hate speech detection and the problem of offensive language," *arXiv:1703.04009v1*, 2017.
- [3] "Hateful users on twitter." <https://github.com/manoelhortaribeiro/HatefulUsersTwitter>. Accessed: 2020-01-23.
- [4] M. H. Ribeiro, P. H. Calais, Y. A. Santos, V. A. Almeida, and W. Meira Jr, "like sheep among wolves': Characterizing hateful users on twitter," *arXiv preprint arXiv:1801.00317*, 2017.
- [5] Y. R. G. F. MacAvaney, Yao, "Hate speech detection: Challenges and solutions," *PLOS ONE*, 2019.