Counterspeech Haikus

Humza Salman {mhs180007@utdallas.edu}

Abstract – In this report we will tackle the problem of generating counter-speech in the form of haikus to hate-speech encountered online. Haikus allow us to incorporate abstract thought into our responses to invoke a deeper meaning by saying fewer words. In our report we consider the generation of 5, 7, 5 Haikus. We utilize state-of-the-art technologies from OpenAI's large language models to develop our pipeline for generating haikus. We will look at some of the underlying reasonings for the pipeline and present the results. Lastly, we will conclude with limitations and present ideas for future work.

I. INTRODUCTION AND BACKGROUND

Encountering hate-speech online typically ruins the browsing experience. Typically, there may be some other comments following in support of or against the hate-speech. The goal of our project is to generate counterspeech in the form of haikus to be more playful and invoke abstract thought. The following will describe our methodology from data collection, to fine-tuning our model, and finally to generating counterspeech in the form of haikus.

GPT-3 – GPT3 is a language model developed by OpenAI which can perform a wide range of NLP tasks including counterspeech. It also contains a family of models available for fine-tuning, such as Davinci, which was released in late 2020. In our pipeline we fine-tune Davinci on the downstream task of generating counterspeech for hate-speech.

Multitarget CONAN – We utilize this dataset which was released in 2021 for the task of generating counterspeech from hate-speech. It is an extension of the CONAN dataset, which is a collection of counterspeech responses to hate-speech written by experts. It contains 5003 entries with varying demographics: Muslims, migrants, women, LGBT+, Jews, POC, disabled, other.

Phonemizer – This is a python library which allows for phonemization of words in the English language. We use the 'festival' backend since it allows for tokenization at the syllable level for English texts.

KeyBert – This is a python library capable of extracting key topics from a document using varying techniques.

text-davinci-003 – This is a GPT 3.5 model which we prompted utilized for Haiku Generation for given topics.

We utilized these technologies in developing our pipeline for generating counterspeech haikus.

II. PIPELINE

For our pipeline we started off by gathering our dataset. We chose Multitarget CONAN because it was released in 2021 and the model we were going to fine-tune, OpenAI's Davinci, was released 2020. We operate under the assumption that Davinci was only trained on CONAN and not the extended dataset we found.

We utilized OpenAI's API to prepare our dataset for fine-tuning and remove any duplicate entries. This allows us to ensure that our data followed their format for fine-tuning their models. It cost us $22.52 USD to fine-tune our model with our dataset. We chose to utilize the first fine-tuned model for generating counterspeech so that we did not increase costs substantially. The cost of fine-tuning jobs is dependent upon the model being fine-tuned and the amount of data being trained on.

To create the rest of our pipeline, we utilized this fine-tuned model to generate counterspeech for any given hate-speech. There are a few hyperparameters to be aware of. The first, *presence penalty (PP)*, ranges from -2 to 2 and it penalizes new tokens based on if they appear in the text so far which helps the model talk about new topics. *Frequency penalty (FP)* ranges from -2 to 2 and it penalizes new tokens based on their existing frequency in the text so far which decreases the model's likelihood to repeat the same line verbatim. *Temperature (T)* controls the degree of creativity of the completion, with higher values being more random and lower values being more deterministic – it ranges from 0 to 1. We also generate *N* samples from each call to the API for completion.

We generate $N_c = 4$, the number of counterspeech candidates, with $T_c = 0.35$, $PP_c = -0.2$, and $FP_c = 0.4$. We also set the maximum number of tokens to 80 to limit our expenditure. Our goal with these values is to

produce creative responses that emphasize talking about the same topics but produce varying sentences.

We take our generated counterspeech responses and our hate-speech prompt and extract their topics using KeyBert to produce $topics_{cs}$ and $topics_{hs}$. For KeyBert, we use Maximal Marginal Relevancy with a diversity of 0.5 to extract unigrams from the generated counterspeech and hate-speech responses. This allows us to produce varying topics from relatively short counterspeech and hate-speech sentences.

We prompt text-davinci-003 with "Write a haiku regarding these topics: $topics_{cs}$" to produce $N_h = 5$ haikus with hyperparameters of $T_h = 0.35$, $PP_h = -0.4$, and $FP_h = 0.3$. We chose these values as they produce diverse haikus in range of the topics given without repeating the same line. However, we came to realize that Haikus generated by Davinci-003 did not always yield the traditional syllable format of 5, 7, 5.

We take all haikus generated by Davinci-003 and create our own synthetic haikus that are in the form 5, 7, 5. We use Phonemizer to accomplish this task. The first line of each haiku should contain 5 syllables, with the second line containing 7 syllables, and the third line containing 5 syllables. We pair the first line of each haiku with the second line of every other haiku, and the third line of every other haiku, and so on to generate our synthetic candidates. We also store any haiku generated by Davinci that follows the traditional haiku format.

Next, we take all our synthetically generated haikus and prune them. In our pruning stage we consider the following: The first line of the haiku should contain a word present within $topics_{cs}$ and $topics_{hs}$. The second and third line of the haiku should each contain a word from either. We do this to ensure that these topics are being utilized in our haikus. This allows us to create synthetic haikus which consider the syllable count of each line while trying to maintain a central theme.

Finally, we output all generated synthetic haikus and haikus generated by Davinci-003 if they match the 5, 7, 5 syllable format.

III. RESULTS

We see that our results vary quite a lot for one simple prompt. We will consider the prompt of "Aliens should not be on Earth and should go back to outer space. They do not belong here." We chose this because it captures the central theme of hate-speech and was not derogatory to any lifeform on Earth. We find that the following results are formed.

Counterspeech generated by fine-tuned model:
1. "On the contrary, they are welcome here, as long as they respect our laws and culture."
2. "I don't see why aliens shouldn't be on Earth. They have the right to be here, just as anyone else."
3. "You are entitled to your opinion, but I think that we should respect the right of others to live and work in the country of their choice."

Davinci Haikus Generated:
"No 5, 7, 5 haikus were generated by Davinci"

Synthetic Haikus Generated:
"Aliens here too,
Culture clashes, hearts in pain;
A battle of wills."

V. CONCLUSION

We saw a pipeline capable of generating counterspeech haikus to a given hate-speech. We chose Haikus as our counterspeech response to be more playful while attempting to provoke abstract thought.

Our pipeline contains no evaluation metrics. It is crucial to evolve this pipeline to consider evaluation metrics at every stage of the process. To note, developing a metric at counterspeech candidate generation could prove to be useful to change hyperparameters to produce diverse counterspeech responses. This would allow us to further diversify the topics we find in $topics_{cs}$.

We should also consider developing a metric for attempting to capture the abstract thought provoked by the haiku. This could potentially be done by generating an open-ended description of the key topics of the Haiku and comparing it to an open-ended description of the key topics of the hate-speech prompt. This would allow us to compare the abstract thoughts of the haiku and the hate-speech prompt to determine if the haiku is properly aligned to be a counterspeech.

Lastly, a better pruning methodology is needed to generate synthetic haikus as ours is naïve. All of this would allow us to consistently generate quality synthetic haikus, which currently require a few iterations or luck.