Classifying and Automatically Neutralizing Hate Speech with Deep Learning Ensembles and Dataset Ensembles

Stanford CS224N Custom Project

Ali Hindy
Symbolic Systems
Stanford University
ahindy@stanford.edu

Varuni Gupta
Department of Computer Science
Stanford University
varuni@stanford.edu

John Ngoi
Department of Computer Science
Stanford University
ngoi.john@stanford.edu

Abstract

Hate speech is one of the most prevalent forms of polarizing language on the planet. This form of human language degrades and disrespects others, yet it is often difficult to detect automatically due to difficulties in understanding language context and bias (oftentimes, directed towards African American dialogue). The invention of social media has amplified hate speech to a magnitude never seen before in human history. To address this issue, we leverage deep ensemble learning techniques to classify and automatically neutralize hate speech. By leveraging the Hugging Face Twitter Hate Speech dataset, our sentiment analysis model is an ensemble system that utilizes a BERT encoder to identify hate speech words and phrases. In addition, we contribute a two-fold pipeline that can detect hate speech given the training samples on a word-by-word basis using a classification model, then replace hateful words with more neutral words using a per-word seq2seq model to generate the neutral word. We ran and evaluated baseline models such as Random Forest, Logistic Regression, Decision Trees, SVC, XGBoost for the classification tasks, yet our HateEnsemble-finetune model outperformed all of them with an F1 score of 99.36%. Human evaluation and our perplexity scores suggest that these data and models are a first step towards the automatic identification and replacement of hate speech in text.

1 Key Information to include

- TA mentor: Manan Rai
- External collaborators, external mentor, sharing project: No

2 Introduction

Hate speech is any kind of communication that attacks or uses pejorative language with reference to an individual or group's religion, ethnicity, nationality, race, or other identifying factor. [1] Hate speech often has more implications than just pejorative verbal language, as it perpetuates intolerance and bigotry and it can potentially lead to violence. For example, the sentence "We want the Arabs out of France" contains hate speech, since there is a derogatory meaning involving wanting an entire group of people out of France due to their identity. We motivate our problem in terms of upholding

Articles 19 and 20 of the United Nations, which provide rights to expression and prevents illegal discrimination. [1] While previous research has focused on hate speech detection, the annotation methods for the datasets used were defined with questionable reliability and inconsistent human classification of hate speech. [2, 3, 4, 5] The goal of our work is to rectify this inconsistency and questionable reliability in annotation by using an ensemble model trained on a diverse amalgamation of datasets. Additionally, we introduce an end-to-end pipeline for text neutralization that suggests edits to replace hate speech with neutral text. The contribution of our paper is as follows:

- We introduced a novel ensemble model consisting of pretrained and finetuned BERT models using an averaged softmax function on our dataset.
- We created a novel dataset ensemble from various datasets to assist in the pretrained BERT models finetuning. The datasets contain text that are labeled as hate-speech and not-hate-speech.
- We have also done some initial analysis to come up with the first end-to-end pipeline for hate speech classification and neutralization, where we suggest edits on a word-to-word basis to replace hate speech with neutral language until the classification model does not recognize the sentence as hate speech

We also provide recommendations for future work in improving our pipeline and suggestions for researchers interested in deep learning ensembles. In the following sections, we will review related work, provide experiments, results, and analysis.

3 Related Work

Since our paper deals with two fields of research, namely Hate Speech Detection, and Text Neutralization, we review relevant literature for each field.

3.1 Hate Speech Detection

Quantitative research in NLP related to identifying hate speech falls under two categories: toxicity prediction and sentiment analysis. Most of the current literature focuses on sentiment analysis, leveraging systems like BERT to analyze text. [6, 4, 7, 8] Additionally, most literature focuses only on text sourced from Twitter, yet these texts use a specific vernacular not present in other forms of writing like in news articles, official documents, speeches, etc. Hence, there are limitations in the data used, as these hate speech detection models cannot generalize beyond tweets.

In recent years, ensemble approaches have been applied to NLP tasks like sentiment analysis for hate speech detection. For example, Hagen et al. introduced a simple ensemble framework for sentiment classification of Tweets. [3] More recently, Zimmerman et al. applied a deep learning ensemble to hate speech classification. Yet, most research focuses on Tweets, likely due to the fact that they are easy to scrape and create large datasets out of. [2] However, these datasets are often skewed, with few samples of hate speech. For example, the dataset used in the Zimmerman et al. paper only contains 5,348 instances of hate speech as opposed to the 13,535 instances of neutral text. This class imbalance in the dataset could be a cause for the poor results of the ensemble method.

We focus on leveraging a deep learning ensemble using BERT models with a balanced dataset (roughly even split of hate speech and neutral text) in order to improve upon prior literature.

3.2 Text Neutralization

Little research has been conducted related to the field of text neutralization using NLP models, yet recently Pryzant, Martinez, and Daas introduced the first model which neutralizes text containing subjective bias. [9] They introduced the WNC (Wiki Neutrality Corpus), as well as the first model to identify subjective bias in text and subsequently fine-tune the generative model. This paper opened up the field for text neutralization using other similar systems and other biases, and Liu and Shibata introduced a similar language generation model that neutralizes abilist text by suggesting neutral replacement for words. [10] Both papers utilize a CONCURRENT system leveraging a generative model, and Liu and Shibata modified this system to employ self-training.

We also leverage a CONCURRENT system as a part of our pipeline, yet we use a deep learning ensemble to identify bias in text as opposed to these two papers. Additionally, we trained our model to identify hate speech as opposed to ableism and subjective bias, and we make our end-to-end pipeline readily available to train on different types of bias and generate more data for this task.

4 Approach

In the era of transformers and Hugging Face, we approached the problem by ensembling pretrained variations of BERT models. The BERT models would output a softmax for the hate speech category, where if the value was closer to 1.0, it would indicate that the sentence or phrase contained hate speech.

As a start, we ensembled dehateBERT and distilroBERTa, and took the average of the softmax output. [11, 12]

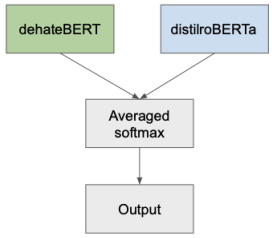

Figure 1: BERT Ensemble

Algorithm 1 BERT Ensemble

Initialize HateEnsemble, passing the models and tokenizers as parameters
Then, in the inference method ...
for Every model **do**
    Encode the inputs using the model tokenizer
    Generate the output using the model and the encoded inputs
    Extract the scores from the output
    Apply softmax on the scores
    Add scores to cumulative softmax scores
end for
Take the average of the cumulative softmax scores
Return averaged softmax score

5 Experiments

5.1 Data

For our experiments, we used the Hugging Face dataset tweets_hate_speech_detection [13] initially and a custom dataset, hatetweets. hatetweets is composed of 4 different hate speech datasets from the corpus made by Davidson et al. in their paper Automated Hate Speech Detection and the Problem of Offensive Language, the UC Berkeley Design Lab, the HSLT Group at Vicomtech, Donostia/San Sebastian, Spain, and the Dipartimento di Informatica, University of Turin. [14, 15, 16, 17]

For the 4 datasets, we normalized the label by making a discrete binary categorization with 0: no-hate-speech, 1: hate-speech. The data preprocessing consisted of converting the categorical hate speech scores to a binary categorization, where the threshold score described by the data collectors as "hate speech" became the threshold for the binary categorization. Additionally, we used conflation to combine the datasets in order to minimize the loss of Shannon information when combining the distributions to create the hatetweets dataset. [18] Conflation is defined for if we have distributions

$P_1, P_2, ..., P_n$ with probability mass functions $p_1, p_2, ..., p_n$, then the combined conflated distribution $\&(P_1, P_2, ..., P_n)$ is continuous with

$$\&(P_1, P_2, ..., P_n) = \frac{\sum_{x \in A} \delta_x \Pi_{i=1}^{n} p_i(x)}{\sum_{y \in A} \Pi_{i=1}^{n} p_i(y)} \quad (1)$$

The distribution of this dataset is shown in Table 1. Below is a summary of the dataset as well as label distribution and word frequency distributions.

The hatetweets dataset contains data from a variety of sources, including tweets, speeches, web forums, and news articles. We intentionally created a diverse linguistic dataset in order to evaluate whether our model could detect hate speech in different scenarios with different diction and rhetoric. Additionally, all types of hate speech, including discrimination based off of age, ability, race, gender, religion, sexuality, and origin are included with a roughly equal split in the dataset.

All 4 of these datasets were unbalanced in terms of labels, with two datasets (Davidson et al., UC Berkeley Design Lab) having a larger proportion of no-hate-speech labels and the other two datasets (HSLT Group at Vicomtech, Dipartimento di Informatica) having a larger proportion of hate-speech labels. However, when combined, these unbalanced dataset proportions summed to a roughly even split of no-hate-speech and hate-speech labels. The hatetweets dataset is one of the only roughly balanced hate speech datasets in existence, and its diverse contents (news, web forums, tweets, etc.) make it a unique and valuable dataset for hate speech classification and neutralization.

We motivate the use of the ensemble dataset as opposed to the non-ensemble dataset for the following reasons: size, balance of labels, and diversity of sources. Although these datasets follow two different distributions (two are skewed right, two are skewed left), the combination of the datasets yields a slightly skewed right distribution that is more balanced than each dataset individually. Furthermore, since we used conflation to combine the datasets, we can be sure that we minimized the loss of Shannon information when combining distributions—yielding a roughly balanced hate speech dataset. [19] We divided all datasets with an 80-10-10 random split with a fixed seed (train-val-test) for our experiments.

Dataset	Number of Samples	μ_{label}	σ_{label}
tweets_hate_speech_detection	31962	0.07	0.26
Davidson et al.	10944	0.13	0.33
UC Berkeley Design Lab	135556	0.35	0.49
HSLT Group at Vicomtech	23353	0.82	0.19
Dipartimento di Informatica	37281	0.91	0.24
hatetweets	**207134**	**0.43**	**0.50**

Table 1: Description datasets in dataset ensemble (hatetweets)

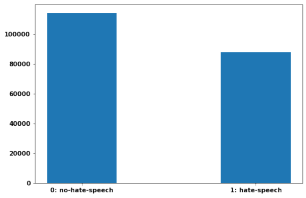

Figure 2: Label distribution in hatetweets training dataset

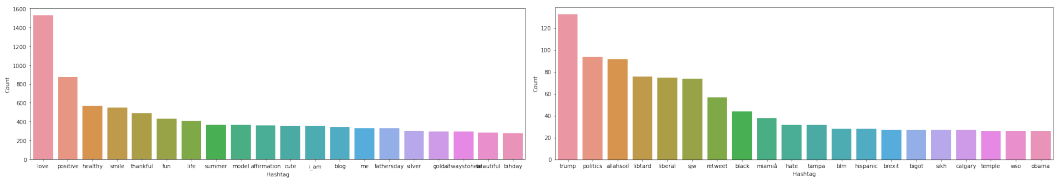

Figure 3: Top 20 most frequently occurring neutral (1) and negative (2) hashtags in hatetweets

5.2 Evaluation method

We used the F1 score to measure the models performance. The reason for picking F1 score as a method of measurement was due to its behavior of being the harmonic mean of the precision and recall and being a great fit for this binary classification task.

We can consider these statements for precision and recall.

Precision: Of all hate speech predictions, how many are actually hate speech?

Recall: Of all actual hate speech sentences, how many are predicted as hate speech?

$$\textbf{F1-score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

We also measure the models Accuracy as a way of giving additional context to the F1 scores and detect any abnormalities in the F1 score numbers.

5.3 Experimental details

We ran baseline evaluations using the RandomForest, Logistic Regression, Decision Trees, SVC and XGBoost models (Code for the Baseline computations is here. For the ensembling, first, we measured the test F1 and Accuracy of the baselines for the dehateBERT, distilroBERTa, and HateEnsemble models. The HateEnsemble-baseline simply ensembled dehateBERT-pretrained-baseline and distilroBERTa-pretrained-baseline models. Next, we finetuned the dehateBERT-pretrained-baseline against the Hugging Face "tweets_hate_speech_detection" training dataset. When we have our the dehateBERT-pretrained-finetune model, we re-initialized the HateEnsemble with the distilroBERTa-pretrained-baseline and dehateBERT-pretrained-finetune and measure the test F1 and Accuracy for the HateEnsemble-finetune. The hyperparameters used for finetuning the models are shown in Table 4. Training time was based on running the training on Google Colab Pro+ with Highest Memory settings.

5.4 Results

The results for the Baseline and the BERT models are shown below in Table 2 and Table 3.

Model	Training Accuracy	Validation Accuracy	F1 Score
RandomForest	0.9992	0.9524	0.6146
Logistic Regression	0.9680	0.9536	0.5901
Decision Trees	0.9992	0.9322	0.5422
SVC	0.9735	0.9538	0.5287
XGBoost	0.9446	0.9433	0.3438

Table 2: Baseline Results of non-NN models

Model	Test F1	Test Accuracy
dehateBERT-pretrained-baseline	0.5543	0.8113
distilroBERTa-pretrained-baseline	0.9729	0.9934
HateEnsemble-baseline	0.9505	0.9865
dehateBERT-pretrained-finetune	0.8900	0.9709
HateEnsemble-finetune	0.9693	0.9916

Table 3: BERT and ensemble models F1 and Accuracy against tweets_hate_speech_detection dataset

For Table 3, HateEnsemble-baseline ensembles dehateBERT-pretrained-baseline and distilroBERTa-pretrained-baseline models. HateEnsemble-finetune ensembles dehateBERT-pretrained-finetune and distilroBERTa-pretrained-baseline models.

Parameter	Value
num of epochs	5
batch size	8
optimizer	AdamW
learning rate	2e-05
weight decay	0.01
training time	$\approx$ 2 hours 24 minutes

Table 4: Ensemble finetuning training hyperparameters against tweets_hate_speech_detection dataset

Even though the Hugging Face tweets_hate_speech_detection dataset is relatively popular, scoring 11.5k downloads, it can be considered a small dataset with only 31,962 which is also split between training, validation and test.

Model	Test F1	Test Accuracy
dehateBERT-pretrained-baseline	0.7468	0.7507
distilroBERTa-pretrained-baseline	0.5621	0.5657
HateEnsemble-baseline	0.6305	0.6359
distilroBERTa-pretrained-finetune	0.9699	0.9703
dehateBERT-pretrained-finetune	0.9369	0.9377
HateEnsemble-finetune	0.9936	0.9937

Table 5: BERT and ensemble models F1 and Accuracy against hatetweets dataset

	Predicted: Hate Speech	Predicted: Not Hate Speech
Actual: Hate Speech	7361	35
Actual: Not Hate Speech	49	6111

Table 6: HateEnsemble-finetune confusion matrix on test set

Parameter	Value
num of epochs	7
batch size	8
optimizer	AdamW
learning rate	2e-05
weight decay	0.01
distilroBERTa training time	$\approx$ 3 hours 47 minutes
dehateBERT training time	$\approx$ 7 hours 22 minutes

Table 7: Ensemble finetuning training hyperparameters against hatetweets dataset

HateEnsemble-baseline ensembles dehateBERT-baseline and distilroBERTa-basline models. HateEnsemble-finetune ensembles dehateBERT-finetune and distilroBERTa-finetune models.

6 Analysis

6.1 Text classification

Initially, we observed an improvement in the HateEnsemble-finetune over the HateEnsemble-baseline. However, we were not satisfied with the experiment since distilroBERTa-pretrained-baseline was trained on the tweets_hate_speech_detection dataset, thus it was likely to perform highly on the tweets_hate_speech_detection test set. So, we evaluated this ensemble on the hatetweets dataset.

We observe that both the dehateBERT-pretrained-finetune and distilroBERTa-pretrained-finetune models performed significantly better after finetune training against the hatetweets train dataset. It

was also comforting to see dehateBERT-pretrained-baseline and distilroBERTa-pretrained-baseline performed poorly against the hatetweets test set. We also observe that the HateEnsemble-finetune model performed the best and reached a very high F1 and accuracy scores against the hatetweets test dataset.

Model	epochs	training time (hours)	Test F1	Test Accuracy
distilroBERTa-pretrained-finetune	3	1:37	0.9535	0.9541
distilroBERTa-pretrained-finetune	5	2:32	0.9679	0.9683
distilroBERTa-pretrained-finetune	7	3:47	0.9699	0.9703
dehateBERT-pretrained-finetune	3	3:57	0.8926	0.8946
dehateBERT-pretrained-finetune	5	5:15	0.9232	0.9242
dehateBERT-pretrained-finetune	7	7:22	0.9369	0.9377

Table 8: Observing scaling law: F1 and Accuracy by training epochs

We also observe that if we train for more epochs, the F1 and Accuracy also increases somewhat linearly and that would elude to the scaling laws. [20]

Given the HateEnsemble-finetune performance is 99.36% F1 Score, we present samples and analysis for the errors that our model made.

For this sample

"FUCK THIS BITCH"

the model predicted that this is not hate speech, whereas the labeled sentence is flagged as hate speech (False Negative (FN)). This false negative could occur because the sentence is all in UPPERCASE, but the training corpus had mix cases and few samples that were all UPPERCASE. One method to prevent this false negative is to preprocess the corpus to be entirely lowercase, or use a BERT uncased model. Next, we consider the sample

"I would eat her pussy till ....."

the model predicted this sample as hate speech, whereas the labeled sentence is flagged as not hate speech (False Positive (FP)). Sexual acts and depiction of sexual parts is typical in hate speech, and the model could have mistakenly predicted this as hate speech. To solve this problem, the models could be finetuned against a corpus that includes these examples as not hate speech.

One fundamental limitation or these models is that they are unable to dynamically determine "who" wrote the sample. For example, if a black person uses the N-word against another black person, it may not always be considered offensive. If a white person uses the N-word in almost any context, that white person would be considered a racist, and the speech that contains the N-word would be deemed hateful. Hence, the determination here of whether or not an utterance is considered hate speech is entirely dependent on the speaker. This nuance of hate speech detection has been explored in prior literature, yet it is important to point out in the context of our paper, since despite our high accuracy on a large dataset, there still exist speaker-dependent limitations to hate speech detection. [21]

6.2 Text neutralization

For Text Neutralization we followed the Pryzant et al. paper which leverages a novel corpus containing 180,000 sentence pairs of subjective bias webscraped from Wikipedia. [9] We ran both models on 10% of our hatetweets dataset (due to compute restraints) and Table 9 shows our results:

Model	BLEU Score
Concurrent Model	0.4743
Modular Model	0.5122

Table 9: Inference Results for Text Neutralization

As steps to run our end-to-end pipeline we implemented the following steps:

1. Tagger Model: The Part Of Speech (POS) tagger model labels the part of speech of each word and tags biased words in the corpus. This process completes in around 2 hours on Google Colab Pro+.
2. Concurrent Model: The Concurrent Model converts biased sentences into neutral form and creates sentence pairs in the form (biased, neutral) using a BERT encoder. This process completes in around 40 hrs in Google Colab Pro+ on only 10% of our data. This step led to a performance bottleneck due to compute resources, as it took the original authors 300+ hours to run this step.
3. Modular Model: The Modular Model contains both Tagger and Concurrent Models. It is a BERT-based classifier to identify hate words and has a novel Join-Embedding through which the classifier can edit the hidden states. We ran it on 10% of our corpus as well.
4. Inference: We ran inference to assess the performance of both the Concurrent and the Modular models. Performance in terms of BLEU scores (score for comparing a candidate translation of text to one or more reference translations) is shown in Table 9.

   We do some concrete results generated after the neutralization part. For example the sentence *"Mark (born 8 march 1964, Watford) is a **disgrace** liberal democrat politician in the United Kingdom , and member of parliament for the Winchester constituency."* gets converted to *"Mark (born 8 march 1964, Watford) is a liberal democrat politician in the United Kingdom , and member of parliament for the Winchester constituency."* with the removal of the "*disgrace*" word.

7 Conclusion

In our experiments, we have proven that the HateEnsemble is able to achieve an F1 score of 99.36% finetuned against our dataset ensemble. Due to the large computing infrastructure required to train the Concurrent Model and our ensemble, we were unable to train our neutralization model on our full dataset, nor were we able to run more experiments on our ensemble model.

With scaling laws, we believe that our ensemble model would perform better if the batch size were larger or if we had the compute to use large pretrained models like the byT5 models. For example, increasing the training batch size from 8 to 16 generates a CUDA out of memory error on a Microsoft Azure NC6s_v3 Ubuntu Linux instance.

Evaluating the qualitative examples can be challenging to review and discuss, as these samples do contain some very hateful content. This also poses a challenge for human turks and deep learning practitioners working on similar tasks, and their willingness to work on these tasks long term due to the emotional damage it may cause.

As good as the intentions are with detecting hate speech and neutralizing them, we also have to be aware that this can also be viewed as a form of censorship and in itself, may be biased on the team or researchers who are evaluating the models performance.

7.1 Future work

For future work, we would like to get access to large compute infrastructure with a sizable budget to be able to train larger models, as well as models with more parameters, bigger batch size and larger balanced datasets. We believe against the scaling laws, that we would achieve state of the art performance with our ensemble approach of pretrained finetuned models.

References

[1] United Nations. United nations strategy and plan of action on hate speech, 2018.

[2] Udo Kruschwitz Steven Zimmerman, Chris Fox. Improving hate speech detection with deep learning ensembles. 2018.

[3] Matthias Hagen, Martin Potthast, Michel Büchner, and Benno Stein. Webis: An ensemble for twitter sentiment detection. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015.

[4] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. *Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion*, 2017.

[5] Ji Ho Park and Pascale Fung. One-step and two-step classification for abusive language detection on twitter. *Proceedings of the First Workshop on Abusive Language Online*, 2017.

[6] Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[7] Georgios Balikas and Massih-Reza Amini. Twise at semeval-2016 task 4: Twitter sentiment classification. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016.

[8] Björn Gambäck and Utpal Kumar Sikdar. Using convolutional neural networks to classify hate-speech. *Proceedings of the First Workshop on Abusive Language Online*, 2017.

[9] Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. Automatically neutralizing subjective bias in text, Dec 2019.

[10] Tiffany Liu and Tyler Shibata. Automatically neutralizing ableist language in text.

[11] Sai Saket Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*, 2020.

[12] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[13] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[14] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515, 2017.

[15] Chris J Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application. *arXiv preprint arXiv:2009.10277*, 2020.

[16] Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[17] E. W. Pamungkas, V. Patti, and V. Basile. Do you really want to hurt me? predicting abusive swearing in social media, 2020.

[18] Theodore Hill. Conflations of probability distributions. *Transactions of the American Mathematical Society*, 363(6):3351–3372, 2011.

[19] Theodore P. Hill and Jack Miller. How to combine independent data sets for the same quantity. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21(3):033102, 2011.

[20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[21] Ashutosh Baheti, Maarten Sap, Alan Ritter, and Mark Riedl. Just say no: Analyzing the stance of neural dialogue generation in offensive contexts. In *EMNLP*, 2021.

8 Appendix

For the project code, see https://github.com/johnnst/cs224n-project.git
Contact johnngoi@stanford.edu for access to the private repository.