CIS 5220 – Final Project – Technical Report

The Neural Networkers

April 2023

Team Members:

- Lavnik Balyan; lavnikb; Email: `lavnikb@seas.upenn.edu`
- Zongxin Cui; zongxinc; Email: `zongxinc@seas.upenn.edu`
- Yash Nakadi; ynakadi; Email: `ynakadi@seas.upenn.edu`
- Eesha Shekhar; eshekhar; Email: `eshekhar@seas.upenn.edu`

Abstract

We perform binary and multi-class classification to identify examples of hate speech and to also differentiate between different kinds of hate speech. We utilise the Implicit Hate Dataset that contains 22,000 tweets that contain no hate, explicit hate, or implicit hate, and the implicit hate examples are further sub-categorised into 6 groups of specific types of implicit hate. We build baseline classifiers (logisitic regression and simple feedforward neural network) and also fine tune the GPT-3 model and the Alpaca model to perform binary classification of hate speech (implicit or explicit) vs. no hate speech and also multi-class classification within the implicit hate speech classes. We obtain good baseline results from the logistic regression model (surprisingly), good results from the fine-tuned GPT-3 model, but unexpectedly bad results from the fine-tuned Alpaca model. Ultimately, these results show that finetuning LLMs could lead to better models for the automated detection and perhaps removal of hateful content from online platforms.

1 Introduction

Despite their many benefits, online platforms that allow people to interact with anyone and post anything they like can can be highly conducive to the communication of toxicity and hate speech. Modern social media platforms struggle with identifying and moderating abusive and hateful comments; when hate speech isn't caught, it can negatively impact people's lives and drive them off the platform.

The goal of our project is to improve upon automatic identification of hate speech, including potentially identifying specific kinds of hate speech. Hate speech identification is a difficult problem in large part because language varies greatly by platform and by target, and because distinguishing between hate speech and other speech containing sensitive words requires a complex understanding of language, including being able to differentiate between comedy and sarcasm, and actual hateful speech. We think that recent developments in natural language processing techniques, specifically the introduction of Large Language Models (LLMs), will allow us to utilise the better understanding of language that LLMs encode in order to improve on existing models. In addition, a complex understanding of language is required for when ”adversarial” hate speech is written to avoid being flagged and moderated. By analyzing the performance of our models across hate-speech categories and fine-tuning it accordingly, we will look to ensure that our model is robust to target variation in its predictions.

2 Related Work

A key flaw in many hate speech detection studies is that offensive language is often mislabelled as hate speech due to an overly broad definition of what constitutes hateful content. In the paper ’Automated Hate Speech Detection and the Problem of Offensive Language’[2], Thomas, Dana, Micheal and Ingmmar trained a multi-class classifier to distinguish offensive languages, hate speech and neither using logistic regression with L2 regularization on a sample of 25k tweets containing hate speech and offensive languages. The best performing model has an overall precision of 0.91, recall of 0.90, F1 score of 0.90 and can differentiate between the three classes. The 31% of true hate speech classified as offensive are genuinely less hateful and were perhaps mislabelled by the coders. It is thus challenging to distinguish the hate speech from offensive language if ignoring the context or if there are no offensive or curse words in the tweets. Error analysis shows that the presence or absence of particular offensive or hateful terms can both help and hinder accurate classification.

The paper ’HateBERT: Retraining BERT for Abusive Language Detection in English’ [3] introduced a re-trained BERT model called HateBERT, which identified abusive language with better results than general BERT models. Absuive language, in the context of this paper, is a category more specific than offensive language but more general than hate speech. In addition to releasing HateBERT and providing results that further proved the effectiveness of this re-training strategy, the paper also released a dataset of English Reddit posts from communities banned for being offensive, abusive, and/or hateful. This dataset, called RAL-E, was used to train HateBERT. To test the effectiveness of finetuning HateBERT as compared to finetuning general BERT, the authors compared the two models on three different (previously existing) datasets: one where any offensive speech is given a positive label and everything else is given a negative label, one where specifically abusive speech only gets the positive label, and one where only hateful speech gets the positive label (OffensEval, AbusEval, and HatEval, specifically). HateBERT outperformed BERT in detecting positive examples and detecting negative examples in each dataset.

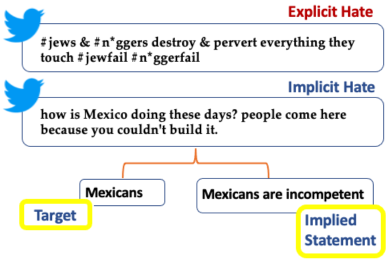
We essentially took inspiration from the above approach and decided to perform fine-tuning on Large Language Models (LLMs) using a specific implicit hate speech dataset because we believe that state-of-the-art LLMs have a better understanding of human language and context clues, and so fine-tuning them could lead to good results. While there is some existing research work in this particular field (finetuning LLMs for hate speech detection), this is still a nascent field. Also, with the advent of bigger and potentially even more powerful LLMs such as GPT-4, the work we do with finetuning GPT-3 could then be applied to better LLMs in the future to achieve even better results (when finetuning of GPT-4 or the other latest LLMs is possible).

3 Dataset and Features

We use the Implicit Hate Dataset introduced in the paper 'Latent Hatred: A Benchmark for Understanding Implicit Hate Speech' [1]. The dataset contains 22,056 tweets from among the most prominent extremist groups in the United States. Of these 22,056 tweets, 6,346 contain implicit hate speech. The dataset is further decomposed into implicit hate classes as follows:

- (24.2%) Grievance: frustration over a minority group's perceived privilege.
- (20.0%) Incitement: implicitly promoting known hate groups and ideologies (e.g. by flaunting in-group power).
- (13.6%) Inferiority: implying some group or person is of lesser value than another.
- (12.6%) Irony: using sarcasm, humor, and satire to demean someone.
- (17.9%) Stereotypes: associating a group with negative attribute using euphemisms, circumlocution, or metaphorical language.
- (10.5%) Threats: making an indirect commitment to attack someone's body, well-being, reputation, liberty, etc.

An example tweet is as follows:



#jews & #n*ggers destroy & pervert everything they touch #jewfail #n*ggerfail

how is Mexico doing these days? people come here because you couldn't build it.

Figure 1: Example of data (ElSherief et al., 2021)

We see that among the kinds of speech (implicit hate, explicit hate, not hate), there is a class imbalance:

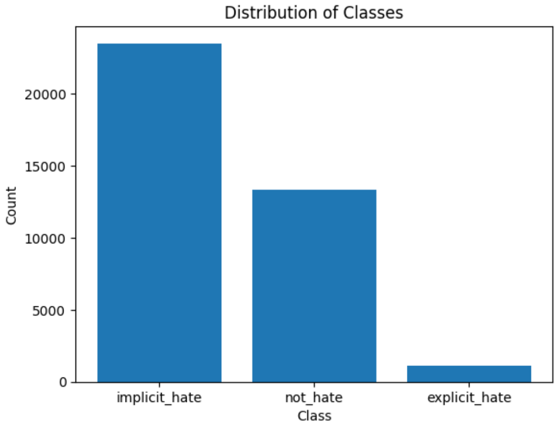


Figure 2: Frequency of class labels

However, when we fold explicit and implicit hate speech examples into one category (yes for hate speech), there is not as much of an imbalance:

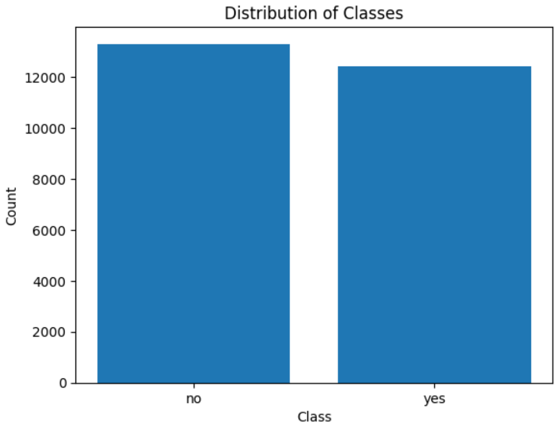


Figure 3: Frequency of yes/no labels

There is also significant class imbalance among the kinds of implicit hate (for example, we have twice as many examples of White grievance as compared to Threatening tweets, and a very small number of tweets in any one sub-category than not-hateful examples overall):

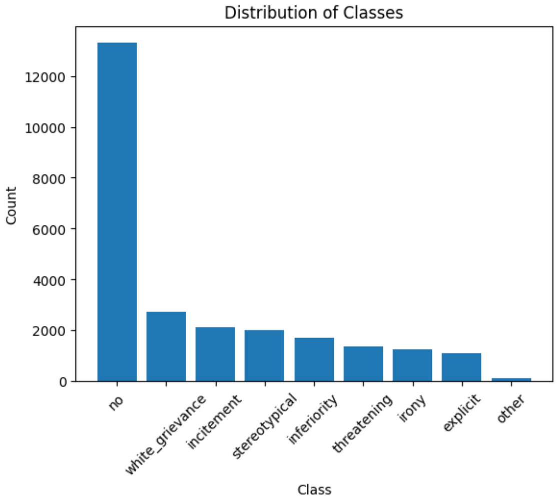


Figure 4: Frequency of implicit hate sub-category labels

4 Methodology

Preprocessing the data for both Baseline Deep Learning and Baseline Non-Deep Learning Methodology:

Since the posts were in text form, it was essential to pre-process the data before the modelling. We processed the data by as follows:

1. Removed the stop words from the text documents: Stop words are commonly occurring words that don't add to the context of the document and it is better to remove them to reduce noise and dimensionality.
2. Converted the data to a matrix of TF-IDF features: In this method, each word is associated with a particular weight to determine how "important" a word is. Furthermore, TF-IDF provides a sparse representation of the data, which helps to reduce the complexity of the data we have.
3. Used a LabelEncoder class to convert the target variable to integer form

Modeling approaches:

1. **Baseline Non-Deep Learning Methodology**:
   - (a) **Binary classification:**
     We used a basic Logistic Regression Model for this purpose, since Logistic Regression is a simple and interpretable model that serves as a good baseline model for tasks involving classification, as seen in Thomas et al. [2].
   - (b) **Multi-class Classification:**
     We used Logistic Regression with the Multinomial Loss Function, using the Stochastic Average Gradient (SAG) Optimization algorithm. We chose this model since it is flexible and can handle the classification of multiple classes. Furthermore, this model is known to perform well on multi-class classification and handles large datasets well.
2. **Baseline Deep Learning Methodology**:
   - (a) **Binary classification:**
     - i. A simple Feedforward Neural network was used for training using the Sequential class from Keras. We used a Feedforward Neural network since it is good at learning complex non-linear relationships in textual data.
     - ii. The model had a single input layer with 64 nodes, a hidden layer with 32 nodes, and finally, an output later with one node and a sigmoid activation function. The Adam optimizer was used and the binary cross entropy loss function was minimized.
     - iii. We trained the model for 20 epochs with a batch size of 32.
   - (b) **Multiclass Classification:**
     This model architecture is very similar to the binary classification one above, as we again used a Feedforward Neural network.

i. There was an input layer with the same length as the maximum length of the padded sequences, a hidden layer with 64 neurons and ReLU activation function, and an output layer with 9 neurons and Softmax activation function.

ii. The Adam optimizer was used and the binary cross entropy loss function was minimized

iii. We trained the model for 20 epochs with a batch size of 32.

3. **LLM Methodology**:
The goal of our research is ultimately to detect hate speech using Large Language Models (LLMs). To achieve this, we prepared two datasets: one for binary classification to determine whether a given post contains hate speech or not, and another for multi-class classification to further categorize the type of hate speech present in the post. Our datasets are carefully labeled to ensure accurate and consistent annotations, and we finetuned the LLMs below using these datasets to improve their ability to accurately identify hate speech.

(a) **GPT3:**
To prepare our data for training, we used the OpenAI data processing tool for pre-processing. This involved removing duplicates, adding end-of-line tokens, adding white space in front of the label token, and splitting the data into training and testing sets. For our base model, we chose GPT-3 Ada due to its efficiency and ability to classify input accurately. We trained the model for 4 epochs with a batch size of 32, a learning rate of 0.1, and a prompt weight loss of 0.01.

To fine-tune the model, we uploaded our datasets to the OpenAI API. After the fine-tuning process was complete, we fed the testing data as input to the fine-tuned model and obtained the results. The binary classification model performed well after fine-tuning. However, the multi-class classification model sometimes only output half a word as a result, such as ”inferior” instead of ”inferiority”. This behavior is understandable and did not produce any results that were not in the training labels. To address this issue in the result calculation, we merged these half words together, ensuring the accuracy of our results.

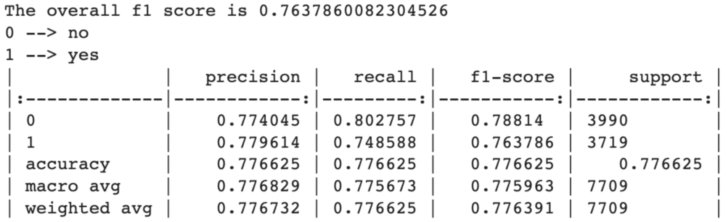
(b) **For Alpaca**
To take advantage of the Alpaca model and meet its training requirements, we added a simple instruction in our prompt that describes the task we are trying to train the model to do. We then set the model parameters as follows: micro-batch size 8, batch size 128, gradient accumulation step 16, epoch 2, learning rate 0.00002, cutoff length 256, LORA R 4, LORA Alpha 16, and LORA dropout 0.05.

5 Results

In this section, we hope to show an improvement in our results as we move from the baseline non-deep learning approach to the baseline deep learning approach, and then to the fine-tuned LLMS.

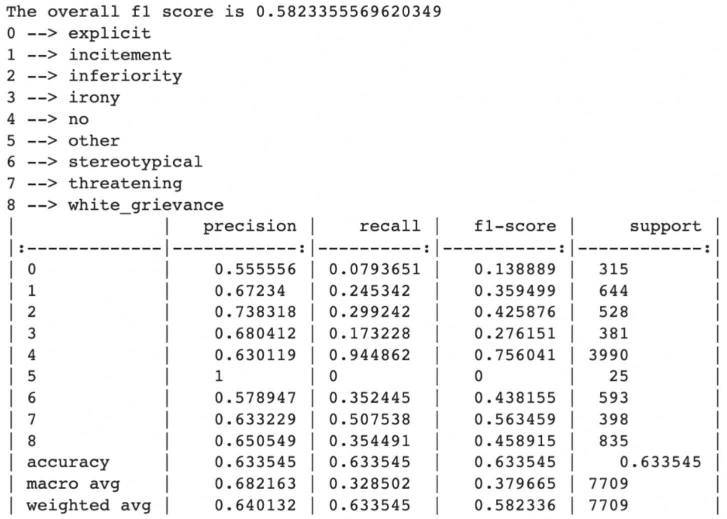
1. **Baseline Non-Deep Learning Results**:

   (a) **Binary classification:**
   The accuracy was **77.66%** with an overall F1 score of **0.763**. The F1 score for 'no' was 0.788 (with a support of 3990) and the F1 score for 0.7637 (with a support of 3719).



```
The overall f1 score is 0.7637860082304526
0 --> no
1 --> yes
|              |   precision |    recall |    f1-score |      support |
|:-------------|------------:|----------:|------------:|-------------:|
| 0            |    0.774045 | 0.802757  |    0.78814  | 3990         |
| 1            |    0.779614 | 0.748588  |    0.763786 | 3719         |
| accuracy     |    0.776625 | 0.776625  |    0.776625 |     0.776625 |
| macro avg    |    0.776829 | 0.775673  |    0.775963 | 7709         |
| weighted avg |    0.776732 | 0.776625  |    0.776391 | 7709         |
```

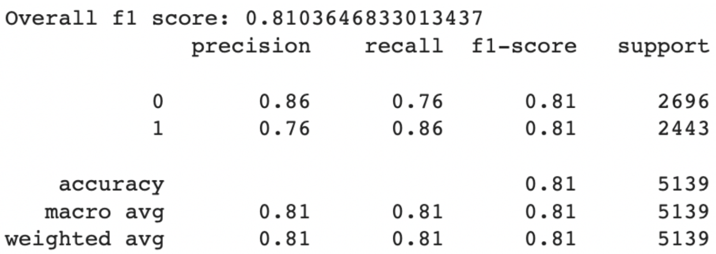
   (b) **Multi-Class classification:**
   The accuracy was **63.35%** with an overall F1 score of **0.5823**.



```
The overall f1 score is 0.5823355569620349
0 --> explicit
1 --> incitement
2 --> inferiority
3 --> irony
4 --> no
5 --> other
6 --> stereotypical
7 --> threatening
8 --> white_grievance
|              |   precision |     recall |    f1-score |      support |
|:-------------|------------:|-----------:|------------:|-------------:|
| 0            |    0.555556 | 0.0793651  |    0.138889 |  315         |
| 1            |    0.67234  | 0.245342   |    0.359499 |  644         |
| 2            |    0.738318 | 0.299242   |    0.425876 |  528         |
| 3            |    0.680412 | 0.173228   |    0.276151 |  381         |
| 4            |    0.630119 | 0.944862   |    0.756041 | 3990         |
| 5            |    1        | 0          |    0        |   25         |
| 6            |    0.578947 | 0.352445   |    0.438155 |  593         |
| 7            |    0.633229 | 0.507538   |    0.563459 |  398         |
| 8            |    0.650549 | 0.354491   |    0.458915 |  835         |
| accuracy     |    0.633545 | 0.633545   |    0.633545 |     0.633545 |
| macro avg    |    0.682163 | 0.328502   |    0.379665 | 7709         |
| weighted avg |    0.640132 | 0.633545   |    0.582336 | 7709         |
```

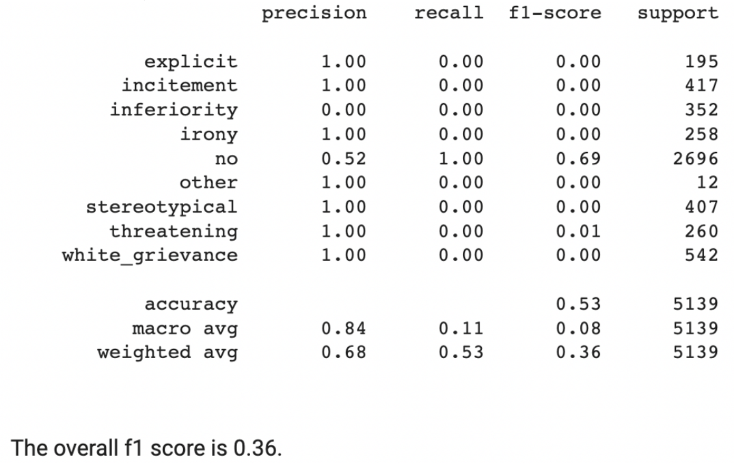
2. **Baseline Deep Learning Results:**

   (a) **Binary classification:**
   The model achieved an accuracy of **80.77%** and an overall F1 score of **0.81**. Both the 'Yes' and 'No' classes had an F1 score of 0.81.



```
Overall f1 score: 0.8103646833013437
              precision    recall  f1-score   support

           0       0.86      0.76      0.81      2696
           1       0.76      0.86      0.81      2443

    accuracy                           0.81      5139
   macro avg       0.81      0.81      0.81      5139
weighted avg       0.81      0.81      0.81      5139
```

(b) **Multi-Class classification:**
The accuracy obtained was **52.5%** with an overall F1 score of **0.36**.



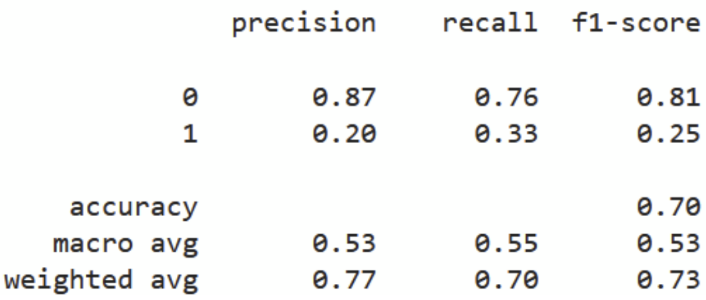
```
                 precision    recall  f1-score   support

       explicit       1.00      0.00      0.00       195
     incitement       1.00      0.00      0.00       417
    inferiority       0.00      0.00      0.00       352
          irony       1.00      0.00      0.00       258
             no       0.52      1.00      0.69      2696
          other       1.00      0.00      0.00        12
  stereotypical       1.00      0.00      0.00       407
    threatening       1.00      0.00      0.01       260
white_grievance       1.00      0.00      0.00       542

       accuracy                           0.53      5139
      macro avg       0.84      0.11      0.08      5139
   weighted avg       0.68      0.53      0.36      5139


The overall f1 score is 0.36.
```

3. **LLM Results**:

When training the models, due to the nature of LLMs, we determined that cross-entropy loss would be the best loss function to minimize.
For the Large Language Models, the results were as follows:

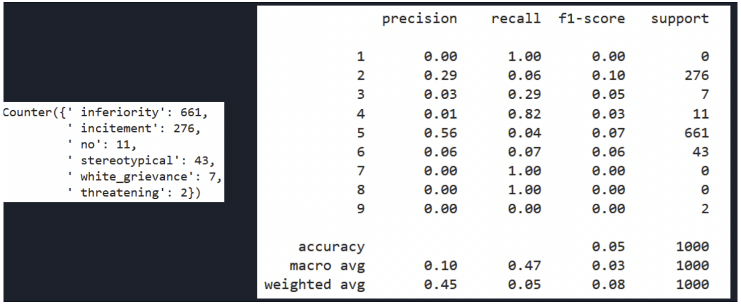
(a) **Alpaca Results**

i. **Binary classification:**
For Alpaca, when testing to see if the model could differentiate between hate speech and non-hate speech, the model had an accuracy of **70%**, with a particular strength (F1 score of **0.81**) at detecting instances of non-hate speech.



```
              precision    recall  f1-score

           0       0.87      0.76      0.81
           1       0.20      0.33      0.25

    accuracy                           0.70
   macro avg       0.53      0.55      0.53
weighted avg       0.77      0.70      0.73
```

ii. **Multi-Class classification:**
However, when trying to differentiate between different implicit hate speech classes, the model performed alarmingly poorly, with accuracy under **10 percent**.

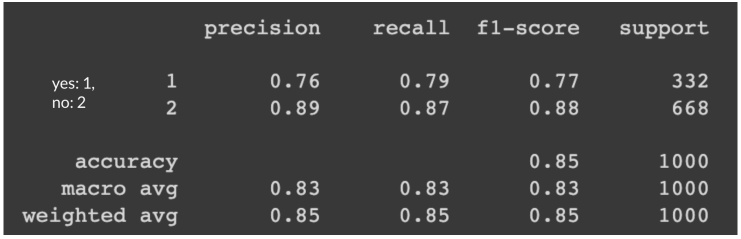


```
Counter({' inferiority': 661,
         ' incitement': 276,
         ' no': 11,
         ' stereotypical': 43,
         ' white_grievance': 7,
         ' threatening': 2})
```

```
              precision    recall  f1-score   support

           1       0.00      1.00      0.00         0
           2       0.29      0.06      0.10       276
           3       0.03      0.29      0.05         7
           4       0.01      0.82      0.03        11
           5       0.56      0.04      0.07       661
           6       0.06      0.07      0.06        43
           7       0.00      1.00      0.00         0
           8       0.00      1.00      0.00         0
           9       0.00      0.00      0.00         2

    accuracy                           0.05      1000
   macro avg       0.10      0.47      0.03      1000
weighted avg       0.45      0.05      0.08      1000
```

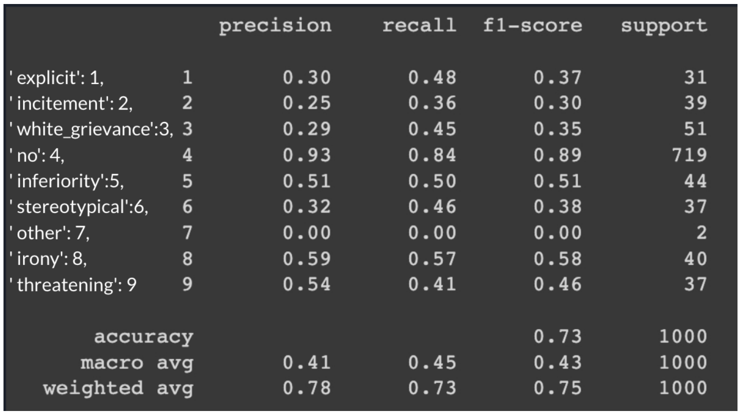
(b) **GPT-3 Results**

i. **Binary classification:**
For GPT-3, however, the results were more promising: For the model with no classes, GPT-3 had a validation accuracy of **85%**, with an F1 score of **0.77** for detecting instances of hate speech and **88 percent** for detecting non-hate speech.



```
              precision    recall  f1-score   support

yes: 1,    1       0.76      0.79      0.77       332
no: 2      2       0.89      0.87      0.88       668

    accuracy                           0.85      1000
   macro avg       0.83      0.83      0.83      1000
weighted avg       0.85      0.85      0.85      1000
```

ii. **Multi-Class classification:**
For the model with implicit classes, GPT-3 had a validation accuracy of **73%**, with an F1 score of **0.89** when detecting instances of non-hate speech



```
                         precision    recall  f1-score   support

'explicit': 1,         1       0.30      0.48      0.37        31
'incitement': 2,       2       0.25      0.36      0.30        39
'white_grievance':3,   3       0.29      0.45      0.35        51
'no': 4,               4       0.93      0.84      0.89       719
'inferiority':5,       5       0.51      0.50      0.51        44
'stereotypical':6,     6       0.32      0.46      0.38        37
'other': 7,            7       0.00      0.00      0.00         2
'irony': 8,            8       0.59      0.57      0.58        40
'threatening': 9       9       0.54      0.41      0.46        37

             accuracy                          0.73      1000
            macro avg       0.41      0.45      0.43      1000
         weighted avg       0.78      0.73      0.75      1000
```

6 Discussion

6.1 Findings

1. **Non-Deep Learning Approach Findings:**

   (a) **Binary classification:**
   The F1 score for ‘no’ was 0.788 (with a support of 3990) and the F1 score for 0.7637 (with a support of 3719). These results are expected as the ’no’ class has slightly higher support. However, it’s worth noting that the accuracy of 77.66% indicates that Logistic Regression possibly struggles with detecting non-linear relationships for the purpose of classification.

   (b) **Multi-Class classification:**
   The accuracy for multi-class classification was lower than the accuracy for binary classification. The F1 score is relatively high overall but less than the previous case (when we weren’t trying to find the specific classes of implicit hate speech). This could stem from the fact that when we are predicting only two classes the classes are more distinguishable than in the case of predicting 9 classes.

   The F1 scores for individual classes is directly related to what its support is, i.e if the support is high it has a relatively higher f1 score. Hence, overall, ”no” has the highest F1 score of 0.847 (it also has the highest support of 3990). Within the different types of hate speech, the highest F1 score is for ”threatening”, which is quite interesting since it only has a support of 398 (which is low compared to the other types of hate speech). The type of hate speech with the lowest non-zero F1 score is explicit hate speech, which has a relatively low support (but not the lowest!) of 0.167. ’other’ was a type of hate speech that had an f1 score of 0, implying that it was never detected in the test set.

2. **Deep Learning Findings:**

   (a) **Binary classification:**
   This model achieved an accuracy of 80.77% and an overall F1 score of 0.81, surpassing the non-DL model. This result was due to the use of a feed-forward neural network, which excels in learning non-linear relationships, as opposed to logistic regression. This is because Feed forward neural networks can capture intricate and complex patterns in the input text data that may not be apparent through traditional statistical methods used in the logistic regression model. Interestingly, both the 'Yes' and 'No' classes had an F1 score of 0.81, despite the 'No' class having a higher support.

   (b) **Multi-Class classification:**
   This accuracy is less than that of the Logistic regression model. The reason why the F1 scores and accuracy is so low for the purpose of classifying the text into 9 different classes could be the fact that Feed-forward neural networks are shallow networks. This means that they have a limited number of layers and may not have enough capacity to handle the increased complexity of a multi-class classification problem. Furthermore, most of the classes were not predicted in the test data, with only "no" and "threatening" being predicted.

   It is noteworthy to notice that the non-DL baseline model gave a satisfactory accuracy using Logistic Regression for both cases and the Feedforward neural network for the first case. It is intriguing especially since TF-IDF rankings were used to pre-process the text data. This was not expected, since TF-IDF rankings rank based on assigning importance to each word. However, in the case of implicit hate speech, the context is not necessarily obvious by assigning importance to individual words.

   One possible explanation is that even though identifying implicit hate speech may require an understanding of context on a deeper level, there may still potentially be identifiable patterns in the language used, and this was accurately learned by the Logistic regression model and Feedforward Neural network.

3. **Alpaca Findings**
   Reasons why Alpaca may not have performed as well as we expected:

   (a) Stronger emphasis on prompt engineering. Unlike the baseline models and GPT-3, we need to provide an instruction to the model, and how we do so may affect the result it generates.

   (b) It constantly classified prompts as "inferiority," meaning we might need to better differentiate the 7 implicit data types, for instance explaining how white greviance is different from inferiority. Again this goes back to chain-of-thought prompting.

(c) Perhaps we need more data. While we used a little under 20,000 samples for the training data, we noticed that there was a degree of class imbalance and a disparity between the number of examples of hate speech and the number that weren't.

(d) Additional finetuning. Of course we can also look to modify hyperparameters like temperature in order to generate more creative responses that might force the model to avoid settling for a specific type of implicit hate speech.

(e) Technical issues. Of course, this technology is extremely new, and so the process of fine-tuning, training and testing the model is not as established and well-understood as other machine learning methods, DL or non-DL.

4. **GPT-3 Findings**
   Why GPT-3 may have performed well:

   (a) Relative to the baseline models, the GPT-3 model was trained in a similar way but compared to a simple FeedForward Network (FFN), a large language model (LLM) is best equipped to handle text data, utilizing embeddings, attention mechanisms, etc. to "understand" the corpus it is provided and thus be capable of understanding our language. This is why we saw improvements despite using the weakest model provided by OpenAI, Ada, at around 350 million parameters.

Our work matters because hate speech detection is still an open research area with deep commercial applications as well. Our results are important because we were able to improve on previous BERT-based models on this Implicit Hate Dataset using the fine-tuned GPT-3 model, and thus we have shown the potential for larger and more capable LLMs to be fine-tuned for this purpose in order to build even better systems.

6.2 Limitations and Ethical Considerations

Limitations

1. **Data Imbalance:** There was a good amount of data imbalance and this could potentially lead to lower F1 scores in certain classes with less support comparatively. For instance, 'other' class had the lowest support and hence, it was often never predicted, as noticed by the F1 scores

2. **Amount of data:** The dataset contained 22,056 tweets, out of which 6346 contained implicit hate speech. More examples of implicit hate speech would allow our model to have higher accuracy in detecting hate speech

Concerns with Alpaca:

We ran into issues with training our Alpaca model, and we believe this was due to a variety of reasons. First, the model defaulted to a singular implicit hate class very often, which could be resolved by optimizing our chain-of-thought prompting to add further context to the types of implicit hate. Second, tuning our hyperparameters to allow for further creativity (eg increasing our temperature parameter) might allow us to see improved results. Third, it would be remiss not to mention that the Alpaca model is very new, and there might have been some technical issues with the training process that will be resolved in the long run as the process becomes more formalized.

Ethical considerations:
A particular consideration is the definition of 'hate speech'. A big part of our project was to focus on implicit kinds of hate speech which are much harder to predict than explicit hate speech. We need to be able to differentiate between comedy/sarcasm and actual hateful content, and this distinction could potentially be an issue requiring further discussion. Our models could be misused if there was tampering with our training data, i.e., if someone listed actual hateful content as non-hateful content and so we would need to prevent data from being messed with through this form of adversarial attack.

6.3 Future Research Directions

The great aspect of this project is that there is an abundance of methods for us to build upon our project. For future steps, we could train more recent (and larger) LLMs (along the lines of GPT-3.5, GPT-4, LLaMA (larger variants), etc.), as well as further finetune our pre-existing models. This further finetuning would require more compute power and monetary resources, but would potentially lead to better results.

An additional point of emphasis would be chain-of-thought prompting, as research has shown that having such prompts leads to improved accuracy.

We could also focus on the data side, perhaps performing this analysis on a variety of datasets and working with larger data in general. We could also scrape data from the internet to make our own datasets in order to ensure a large enough size.

7 Conclusions

Our project aimed to identify the most effective model for detecting implicit hate speech. This was a significant challenge, as detecting this particular type of hate speech requires a deep understanding of the context and cannot be determined solely on the basis of individual words such as slurs. We also wanted to see whether the models can accurately detect the specific type of hate speech involved. We looked at several models, including non-deep learning models such as logistic regression, deep learning models like feed forward neural networks, language models such as GPT-3 and Alpaca, and fine-tuning of the language models.

Looking at all the models, it is clear that each model had a higher accuracy and F1 score when attempting to do binary classification as compared to Multi-Class classification.

The Non Deep Learning model performed better than the Deep Learning Model in Multi Class Classification while the Deep Learning Model performed better in Binary Classification.

The Non Deep Learning model performed better than finetuned Alpaca in both binary classification and multi-class classification

Overall, we found that finetuning GPT-3 provided the highest accuracy and F1 scores for both binary and multi-class classification. Therefore, we can conclude that finetuning GPT-3 is the most effective way to detect implicit hate speech among our current approaches.

This lends further credence to our belief that this kind of finetuning work applied to the latest and bigger LLMs could lead to even better results than what we achieved with GPT-3.

8 References

[1] ElSherief, M., Ziems, C., Muchlinski, D., Anupindi, V., Seybolt, J., De Choudhury, M., & Yang, D. (2021). Latent Hatred: A Benchmark for Understanding Implicit Hate Speech. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)

[2] Davidson, T., Warmsley, D., Macy, M., & Weber, I., "Automated Hate Speech Detection and the Problem of Offensive Language" (2017, March 11). arXiv doi: 10.48550/arXiv.1703.04009

[3] Tommaso C., Valerio B., Jelena M., & Michael G. (2021, February 4). "HateBERT: Retraining BERT for Abusive Language Detection in English". arXiv:2010.12472 [cs.CL].https://arxiv.org/abs/2010.12472