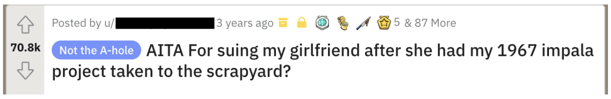
Are you the A**hole?

Albert Hung, Raef Maroof, Colton Peffer, Kate Shenton
{azhung, maroofr, cpeffer, kshenton}@umich.edu

1 Introduction

What goes into determining if someone does something good or bad? What factors influence our judgements on someone's actions? Do we judge people and scenarios by a universal sense of morality, or do other factors influence our judgements? Human morality is a complex and nuanced concept, with people often judging others' actions based on factors beyond the simple binary of good and bad. In this paper, we explore these judgments within the context of the Reddit community r/AmItheAsshole, where users can post scenarios from their lives for others to assess if they were the "asshole" or not. Figure 1 shows an example post, that shows the post title and the post classification by the subreddit community. **We hypothesize that the final label assigned to a post is dependent on the post's text, sentiment, and topic. Our goal was to create a model that can predict whether a post will be labeled as "asshole" or "not the asshole" based on these factors.**



Posted by u/[redacted] 3 years ago 5 & 87 More

70.8k

Not the A-hole AITA For suing my girlfriend after she had my 1967 impala project taken to the scrapyard?

Figure 1: r/AmItheAsshole post example

Previous research has explored the r/AITA subreddit using interpretable models to classify posts; however these classifiers used features outside the post such as comments and user karma; there is still a gap in the literature when it comes to analyzing community judgments using solely post textual data.

The problem we address is twofold: first, we need to understand the linguistic patterns that contribute to the community's judgment, and second, we must develop an accurate and interpretable machine learning model that can predict these judgments.

To address the first goal we build up a dataset of additional features characterizing each post including Doc2Vec, ngrams, LIWC, emotion, and topic. We then use the interpretable machine learning models Logistic Regression, Random Forest and Naive Bayes in order to gain insights into the relationships between the additional features and the classification.

With logistic regression an ablation study was performed, iterating through groups of features. This model did perform the highest among the interpretable models with the feature sets {top 1000 trigrams} and {Unigram + LIWC + Emotion}. We further explored connections with features through a random forest classifier. We predicted random forest would perform well with our high dimensional dataset, however the accuracy was lower than logistic regression. We did gain insight into features of high importance to the classifications. Doc2Vec features were the most important in classification, although they were not highly interpretable. The features present in a post that slightly correlated with the label 'Not the Asshole' included TopicProb0 (friends), Dic (Percent words captured by LIWC), and the ngrams 'edit', 'she', and 'her'.

To strengthen our model, deep learning models were explored, including the Neural Networks Text CNN, Bidirectional GRU Network, and an Ensemble Classifier. The Bidirectional GRU Network achieved the highest results, matching that of logistic regression indicating context is of the highest important when classifying posts as 'asshole' or 'not the asshole'. The pretrained model BERT, which captures a large degree of context, unsurprisingly achieved the highest classification balanced accuracy.

In summary we found there was no one feature set that strongly correlated to a label of 'asshole' or 'not the asshole'.

The remainder of this paper is organized as follows: Section 2 presents related work; Section 3 explains our data and feature extraction; Section 4 details the algorithms and methodology used; Section 5 discusses our results; Section 6 explains ethical considerations; and Section 7 concludes the paper and suggests directions for future work.

2 Related Work

Several recent studies have explored the r/AITA subreddit to both create a model that is be able to predict the outcome as well as try to identify the features that contribute to the model's success. Most of the previous work classified r/AITA comments with high accuracy, but not the actual posts. One recent work has explored using BERT to classify the actual posts, achieving a 62 percent accuracy. When evaluating the influences of specific words, very strong correlations were not found between word choice and subreddit outcome [2].

Another study tried more interpretable models, including naive bayes, decision trees, k-nearest neighbors, logistic regression, neural networks, random forests, and SVM, achieving the best performance with random forests with a 78 percent accuracy. However, the final model was heavily reliant upon social features, such as author karma [3]. We want to explore the impact of just the text on the community decision, with no additional information. There also has been some work done in terms of identifying unigrams that are correlated with each outcome; however, machine learning models were not used to leverage the learned information [4].

To our knowledge, there are no studies that are trying to identify optimal text features using machine learning to predict the outcome of posts. Furthermore, Roberta has not been utilized on this specific subreddit.

3 Data

3.1 Data Collection

In order to evaluate the posed research question, a dataset of r/AmItheAsshole posts compiled by Elle O'Brien was used [5]. These were collected from the forum dating from the first post in 2012 to January 1, 2020. In total, there are 97,628 samples with base features: id, timestamp, edited, verdict, score, number of comments, is_asshole. Verdict is the exact designation of the post between four different classes but for simplicity we focus on is_asshole, which is binary between 0 - not the asshole and 1 - asshole, based on verdict. This will be the label that was trained on. It is important to note that the data is imbalanced with about 72% being the ”not the asshole” and 28% being ”the asshole”. This imbalance will be accounted for in the models on a case by case basis.

3.2 Data Preprocessing

The first thing data processing step was to concatenate the title and body so all our contextual information was contained in one feature. From here, standard text preprocessing was performed, such as casting to lower case, removing stop words, POS tagging, stemming, and lemmatization. Pronouns were left in because we believed this could potentially remove vital contextual information that the model could use.

3.3 Additional Features

From here, additional features were created based on the processed text using outside models in the hopes of learning complex information encoded in the text such as topic, emotion, authenticity, etc. These new features will be incorporated in the logistical regression model to see if they provide additional information about the label.

Unigram, bigram, and trigram tf-idf vectors, emotion vectors using RoBERTa-base, Doc2Vec embeddings, topic vectors using Genism LDA, and LIWC features were created.

The emotion vectors consist of 4 feature probabilities between the labels Joy, Optimism, Sadness, and Anger. Doc2Vec converts text into n-dimensional feature representations by representing word content similarity. Based on experimentation, 150 dimensions has the best performance. The topic vectors produce a probability of a given text being in a topic identified via LDA clustering. The topics are Work, Family, Home, Driving, and Friends. LIWC uses predefined dictionaries that provides a score for a variety of categories (analytic, authenticity, etc.) and part of word usages. This produces a vector with 118 dimensions. With these features in hand, the base features (processed text, word count, is_asshole, etc.) and additional features have all been collected.

4 Methods and Algorithms

All models used stratified 5-fold cross validations to account for the class imbalance in the dataset, and unweighted average recall (UAR) was used to

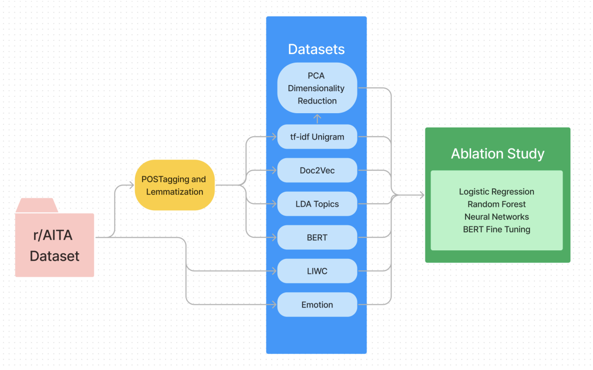


Figure 2: Dataseet Preprocessing and Training Flow

evaluate each model.

4.1 Logistic Regression

Logistic regression is a commonly used, easily interpretable machine learning model that takes a linear combination of the input features and passes it through a sigmoid activation function to arrive at an output. An elastic net penalty was chosen for the regularization term and a grid search was used to find the optimal hyperparameters. The hyperparameters explored included C (penalty term) [0.01, 0.1, 1, 10, 100], l1_ratio (elastic net mixing parameter) [0, 0.2, 0.4, 0.6, 0.8, 1].

To find the optimal groups of features, an ablation study was performed on all features. Each set of features were tested individually, including unigrams, bigrams, and trigrams, Doc2Vec embeddings, LIWC, emotion, and topics. Principal component analysis (PCA) was also performed in an attempt to minimize the dimensionality of text embeddings to 50 and 100 components. After finding the optimal text features, all of the non text features were added separately to determine the most impactful group of features. This process was then repeated one additional time.

To find the most optimal individual features, minimum redundancy - maximum relevance (mrmr) feature selection was used. Then, in order of most influential features, correlated features with a correlation of over 0.4 were removed. Finally, features that were statistically significant ($p < 0.01$) were chosen for the final feature set.

4.2 Random Forest

We used another interpretable model, Random Forest, to further explore notable features in our dataset used in classification. We chose Random forest as it can be effective at handling high-dimensional data and provide robust results with reduced risk of overfitting. This is attributed to its use of multiple decision trees, random sampling of training data, and random selection of a feature subset for each tree.

To optimize the model, a grid search was performed while first considering all available features as inputs. The hyperparameters explored included n_estimators (number of trees) [50, 100, 150], max_depth [10, 20, 30], max_features [sqrt, log2, None], min_samples_split [2, 5, 10], and min_samples_leaf [1, 2, 4].

Although random forests are relatively insensitive to differences in scale between features, input features were standard scaled for accuracy. Additionally, the class weight parameter was set to ”balanced” to account for the unbalanced data set between labels. After the model was fit, the top 10 features importances were graphed, and the balanced accuracy of the model was noted.

To further improve the model's accuracy, recursive feature elimination (RFE) was employed with the random forest classifier to decrease the dimensionality of features in the final classifier. RFE functions by fitting a model with all features, calculating the feature importances, eliminating the least important feature, and continuing the process until the desired number of features is reached. In this case, the number of features extracted was 100.

4.3 Neural Networks

We chose to explore three key architectures for Neural Networks: a Text CNN, a Bidirectional GRU Network, and an Ensemble Classifier. For all neural networks, we represent posts as series of 300 dimensional GloVe word embeddings of varying length. To limit our search space, we only explore text features.

Our Text CNN uses filter sizes of 3, 4, and 5. After the convolutional layer, we obtain a feature mapping that is fed into a hidden layer with 256 nodes and ReLu activation before the output layer, which has sigmoid activation. An illustration of this is provided in Figure 3

Our Bidirectional GRU Network consisted of a single bidirectional GRU with a hidden size of 300 into which the GloVe embeddings of each word in a post is fed one at a time. The outputs of the Bidirectional GRU across all timesteps are max pooled to generate a single embedding representing the entire post with length 600. This embedding is

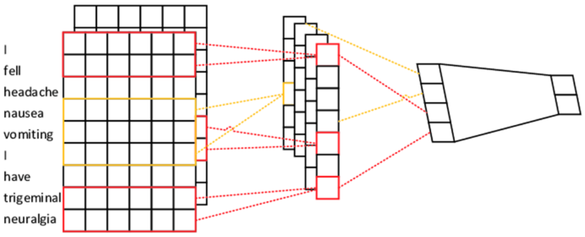


Figure 3: Text CNN Architecture [1]

passed into a feed forward neural network with one hidden layer, using 512 nodes and tanh activation, and an output layer using sigmoid activation. An illustration is provided in Figure 4.

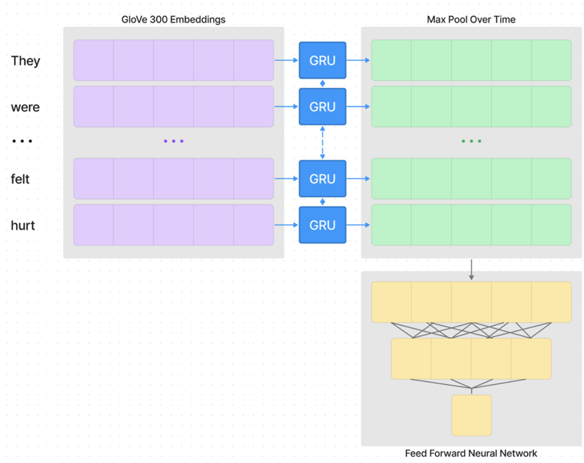


Figure 4: Bidirectional GRU Architecture

Lastly, our Ensemble Classifier was trained to intuitively learn a weighted average of outputs from the Text CNN and Bidirectional GRU Network. Outputs from these two classifiers are fed into a layer with 2 nodes and linear activation before the output layer, once again with sigmoid activation.

When training all classifiers, we used the Adam optimizer with Binary Cross Entropy loss and UAR based early stopping for a maximum train time of 100 epochs. For simplicity, we only evaluate dropout rates in the range $\{0, 0.1, 0.25\}$ and learning rates in the range $\{0.01, 0.005, 0.001\}$ under a fixed batch size of 64. Additionally, given our class imbalance, we explored weighting our positive samples approximately 2.5 times more than negative samples.

4.4 BERT Fine-Tuning

For the second attempt at using a non-interpretable deep learning model, we chose Google BERT (Bidirectional Encoder Representations from Transformers) as it performs very well at maintaining context from text to vector representation. We wanted a model that hopefully better took into account the context contained within the text in a way that did not have to be interpretable. BERT is a large pretrained transformer based model. Specifically, we used the RoBERTa configuration, which builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates [6]. This results in better overall performance when compared to base BERT on human-level NLP [7].

Our model takes the preprocessed text and sends it to BERT, which outputs a vector with 768 dimensions. From here, we implement the ”fine-tuning” of BERT which in our case essentially means applying some sort of deep learning technique to this vector. We applied a single layer neural network with dropout. Additionally, we used cross entropy loss with initialized weights to account for the class imbalances and PyTorch’s Adam optimizer. Both of these were chosen as they are shown to perform well on binary classification and for a single layer neural network. A number of hyperparameters can be tuned in our model including but not limited to: train and validation batch size, number of epochs, learning rate, and dropout rate. Due to the long training time of the model, epochs were limited to three while other parameters were tuned. Based on validation loss versus training loss, the model had yet to overfit and if anything was underfit by the small number of epochs.

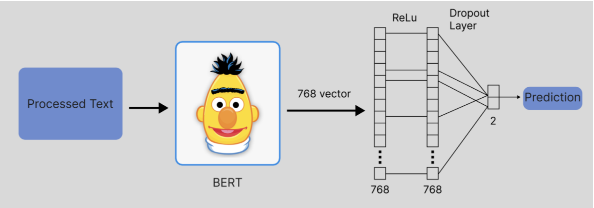
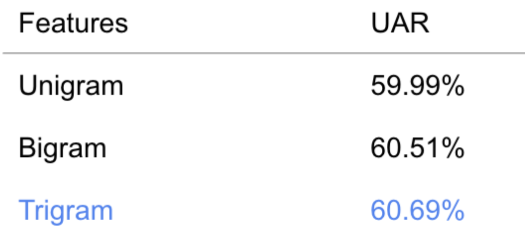


Figure 5: BERT Fine-Tuning Architecture

5 Results and Discussion

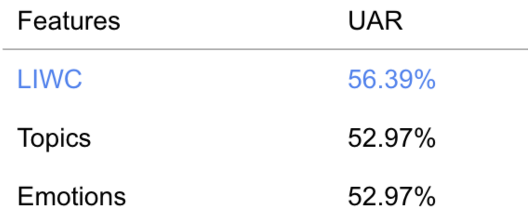
5.1 Logistic Regression

The best performing text features were trigrams, as shown in Figure 6, suggesting that the more context that is added, the better the model will perform. Furthermore, the Doc2Vec and PCA embeddings perform poorly, suggesting that both of these methods are unable to capture meaningful information to the task at hand. The best performing non-text



Features	UAR
Unigram	59.99%
Bigram	60.51%
Trigram	60.69%

Figure 6: Text feature accuracy



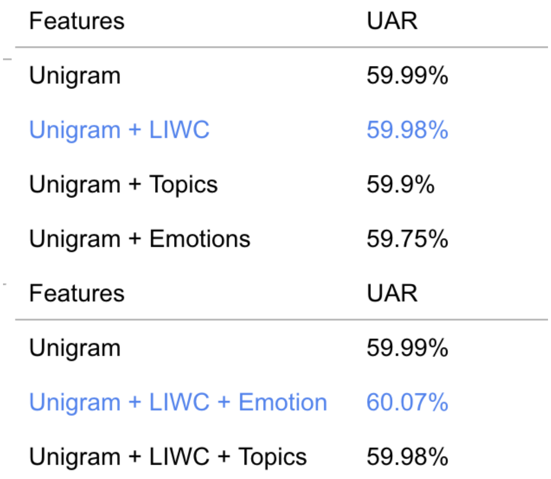
Features	UAR
LIWC	56.39%
Topics	52.97%
Emotions	52.97%

Figure 7: Non-text feature accuracy

features were the LIWC features, as seen in Figure 7, which implies that it provides more information pertinent to the task than emotions and topics but not as much information as the source text.

Moving forward with the ablation study, only unigram text features were explored since the difference in accuracies between the trigram features and unigram features were minimal, and the dimensionality of the trigram features was double that of the unigram features, making trigram features computationally expensive. After adding one non-text feature to the unigram features, the accuracy slightly decreases, as seen in Figure 8, which hints that LIWC features by themselves act as noise when combined with unigram features. This may be due to the drastic differences in dimensionality between these two representations (unigram - 1683 features; LIWC 120 features). One more non-text feature was added to the unigram and LIWC features to conclude the ablation study. The best accuracy was achieved by unigram, LIWC, and emotions, even outperforming the unigram features.

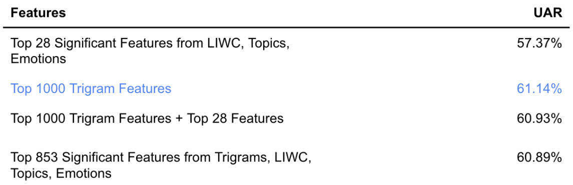
For the best individual features, the results are shown in Figure 9. The accuracy of the best trigram features performed better than the best combined features (trigram + all non-text features), suggesting that the most information can be gained just from the text.



Features	UAR
Unigram	59.99%
Unigram + LIWC	59.98%
Unigram + Topics	59.9%
Unigram + Emotions	59.75%

Features	UAR
Unigram	59.99%
Unigram + LIWC + Emotion	60.07%
Unigram + LIWC + Topics	59.98%

Figure 8: Results from adding up to two non-text features to unigram features

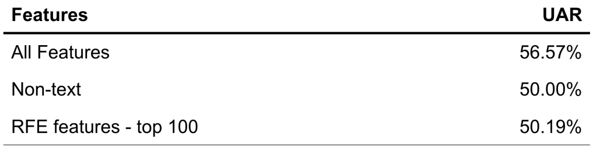


Features	UAR
Top 28 Significant Features from LIWC, Topics, Emotions	57.37%
Top 1000 Trigram Features	61.14%
Top 1000 Trigram Features + Top 28 Features	60.93%
Top 853 Significant Features from Trigrams, LIWC, Topics, Emotions	60.89%

Figure 9: Accuracy of Top Features

5.2 Random Forest

The results of Random Forest classification are shown in Figure 10. Through performing grid search, the classifier achieved the best results through 100 trees with a max depth of 10. Increasing these hyperparameters led to decreased balanced accuracy and F1 score, suggesting overfitting.



Features	UAR
All Features	56.57%
Non-text	50.00%
RFE features - top 100	50.19%

Figure 10: Results of a Random Forest Classifier on different feature sets

Inputting all features into the classifier achieved the best results at 56.67%, and the top 10 feature importances in this classification are shown in 11. The Doc2Vec features are the most important features in this classification, however they are not particularly interpretable. The other features of high importance include Authentic (Perceived honesty, genuineness), Dic (Percent words captured by

LIWC), and TopicProb0 (friends).

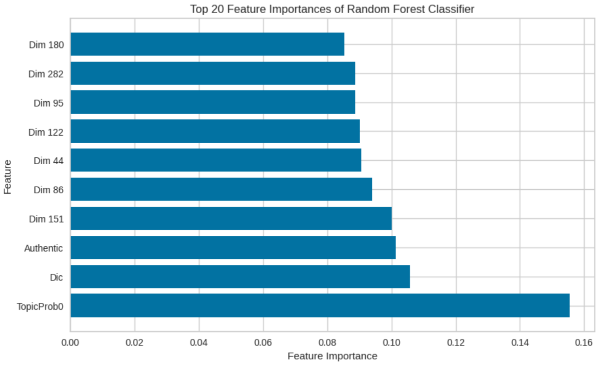


Figure 11: Top 10 feature importances used in random forest classification with all features

There was a substantial drop in balanced acccuracy when features were reduced through recursive feature elimination, shown in Figure 10.

We were able to use the top features presented through random forest classifiers and RFE to further exam feature correlations to the is_asshole label. The top ngrams and nontext features extracted through RFE are represented in Figure 12 and Figure 13. Figure 12 represents the correlations between the top five importances of specific unigrams, bigrams, and trigrams. This indicates there is a slight correlation between the use of the words 'edit', 'she', 'her', and 'friend,' with a label of 'Not the Asshole,' while the use of the words 'time', 'feel', 'want', and 'my mom,' being present in the top ngrams indicates a slight correlation to being labeled 'Asshole.' These findings are in line with previous literature as explored in [4]. The word 'edit' is connected to how users react after a label is given rather than a feature of a post pre-judgement, so in future work should be filtered during preprocessing. The demographic words 'she', 'her', and 'my mom' could indicate a slight bias in judgement or could instead be connected to the context of the post, such as certain demographics writing about less morally ambiguous situations. This could only be determined with a much more granular anlysis of context, and cannot be determined with current information. Figure 13 shows a slight correlation with a post topic centered around 'friends' being labeled as 'Not the Asshole.'

An attempt was made to use RFE with five-fold cross-validation to find the optimal number of features. However, due to the computationally intensive nature of this approach, it timed out in several attempts. Consequently, optimizing the number of features using cross-validation remains an area for future work.

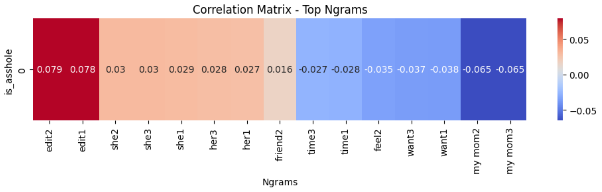


Figure 12: Correlation between top ngram features and label "Not the Asshole"

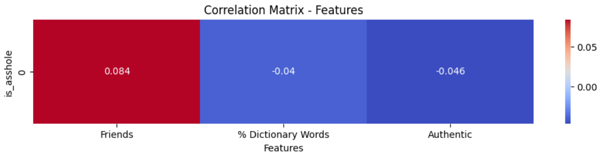


Figure 13: Correlation between top non-text features and label "Not the Asshole"

5.3 Neural Networks

We trained our neural networks using specialized computing infrastructure due to long training times. The results are shown in Figure 14.

Model Type	**UAR**
Text CNN	59.94%
Bidirectional GRU Network	61.14%
Ensemble Classifier	60.61%

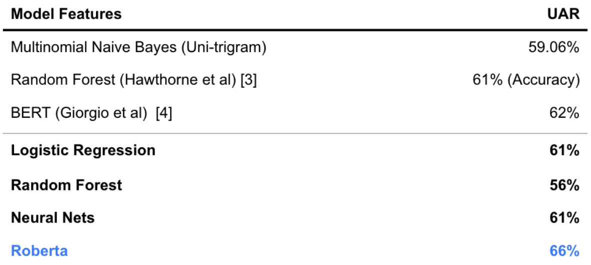
Figure 14: Results of training Neural Network Architectures

We found the the Bidirectional GRU network outperformed other architectures and manages to match our highest logistic regression UAR. Interestingly, it also outperforms the ensemble classifier, which may be expected to at least match the highest performing of its two constituent classifiers.

Intuitively, the Text CNN should be able to capture and prioritize key features without drowning their influence in long context windows. At the same time, it can only capture text spans of at most 5 words. In contrast, the Bidirectional GRU Network can, in theory, capture arbitrarily long spans of text. Given that the Bidirectional GRU had the highest performance, we may infer that context is of the highest important when classifying posts as 'asshole' or 'not the asshole', so much so that considering Text CNN outputs does not help performance in the Ensemble Classifier. This makes sense, given the oftentimes nuanced nature of moral judgements.

5.4 BERT Fine-Tuning

The best testing UAR we were able to achieve through our BERT model is shown below in Figure 15 against both baselines and other approaches taken. The UAR performance just below 66% which outperformed every other model. It performed the best with the following hyperparameters: training batch size 16, validation batch size 4, learning rate .001, and dropout .15.



Model Features	UAR
Multinomial Naive Bayes (Uni-trigram)	59.06%
Random Forest (Hawthorne et al) [3]	61% (Accuracy)
BERT (Giorgio et al) [4]	62%
Logistic Regression	**61%**
Random Forest	**56%**
Neural Nets	**61%**
Roberta	**66%**

Figure 15: Results Compared

It is interesting that BERT was able to perform the best as it shows a couple things. It lends credibility to our previous discoveries that there are no features which are heavily correlated with the label. BERT outperforming all the others shows that the determination of whether a post is given the ”asshole” designation most depends on the model that best considers complex language structure and meaning. Understanding what is being said with context is key to predicting the right label. Intuitively, this is a no-brainer but it is still interesting to see that BERT outperformed the other models and encourages future work focused exclusively on text transformation.

6 Ethical Implications

One ethical consideration when using Reddit post data from r/AmItheAsshole is that the data may not be representative of the broader population. As the posts are from a specific subreddit, the model's predictions may be biased towards the views and judgments of that particular community. To address this, we interpret our results with respect to the Reddit population without extending them to a more general context. A classifier trained on this specific dataset should not be used to make moral judgments for any purpose outside of this context.

Another ethical concern is the potential compromise of users' privacy and anonymity. Although the dataset does not include usernames, the posts may still contain information that could reveal a user's identity. To mitigate these concerns, we will present our findings in a way that does not disclose any identifying information. This will help ensure the privacy of users who have shared their experiences on the subreddit.

Using BERT to improve classification performance may raise ethical challenges as it is a non-interpretable model, and thus the reasons for each classification are not available for analysis. Biases in the model will be much harder to recognize and the model should not be deployed without more rigorous testing.

Although possible features that influence whether a poster is labeled at fault or not were detected, such as specific gendered n-grams and topics, we discuss the possible reasons behind these findings. However, no overarching conclusions should be extrapolated from the limited data and evidence. It is essential to recognize the limitations of the dataset and the model's predictive capabilities when interpreting and applying the results.

7 Conclusion

While we were able to reach a higher performance than approaches currently found in literature that operate only on text data, it is of note that we could not find a feature or set of features from our dataset that strongly influenced the moral decision of the story. In fact, as discussed in our results, all correlations between features and our label were extremely low. Unsurprisingly, large language models outperform all interpretable models. Furthermore, models that take into account context (including LLMs, Bidirectional GRUs, etc) also perform quite well. Critically, however, these models all lack interpretability. Given our original goal of understanding exactly which factors influence judgements, feature engineering and selection with interpretable models that consider longer spans of text may be key. Additionally, as found in our exploration of logistic regression and random forest, introducing variables quantifying certain overarching aspects of posts like topic or speaker authenticity (and other LIWC related metrics) seems to help gain a greater understanding of judgements. It may also be fruitful to explore factors outside of a post's content, such as the subreddit membership of the speaker and different commenters on a post. Such metrics may allow further insight into the biases with which a post is created and viewed.

Our code can be accessed at https://github.com/RaefM/eecs448-mde.

8 References

[1] C. Yao, Y. Qu, B. Jin, L. Guo, C. Li, W. Cui, and L. Feng, "A convolutional neural network model for online medical guidance," IEEE Access, vol. 4, pp. 4094–4103, 2016.

[2] Efstathiadis, I. S., Paulino-Passos, G., & Toni, F. (2022). Explainable patterns for distinction and prediction of moral judgement on reddit. arXiv preprint arXiv:2201.11155.

[3] Haworth, E., Grover, T., Langston, J., Patel, A., West, J., & Williams, A. C. (2021). Classifying Reasonability in Retellings of Personal Events Shared on Social Media: A Preliminary Case Study with /r/AmITheAsshole. Proceedings of the International AAAI Conference on Web and Social Media, 15(1), 1075-1079. https://doi.org/10.1609/icwsm.v15i1.18133

[4] Giorgi, Salvatore & Zhao, Ke & Feng, Alexander & Martin, Lara. (2023). Author as Character and Narrator: Deconstructing Personal Narratives from the r/AmITheAsshole Reddit Community. 10.48550/arXiv.2301.08104.

[5] O'Brien, Elle. (2020). AITA for making this? A public dataset of Reddit posts about moral dilemmas.

[6] Liu, Y. & Ott, M. & Goyal, N. & Du, J. & Joshi, M. & Chen, D. & Levy, O. & Lewis, M. & Zettlemoyer, L. & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692

[7] Ganesan AV, Matero M, Ravula AR, Vu H, Schwartz HA. Empirical Evaluation of Pretrained Transformers for Human-Level NLP: The Role of Sample Size and Dimensionality. Proc Conf. 2021 Jun;2021:4515-4532. doi: 10.18653/v1/2021.naacl-main.357. PMID: 34296226; PMCID: PMC8294338.