Who do we call assholes? Classifying Moral Judgement using Reddit Data from r/AmITheAsshole

Reaction Paper and Project Proposal

Kirara Kura
Department of Statistics
University of Washington
Seattle, WA, 98105
kurak25@uw.edu

Wenhao Pan
Department of Statistics
University of Washington
Seattle, WA, 98105
wenhaop@uw.edu

Garrett Allen
Department of Statistics
University of Washington
Seattle, WA, 98105
gpa5@uw.edu

1 Introduction

As moral actors, it is common that we must ask ourselves if we are acting in ways that most other people would deem reasonable or moral. Ordinarily, one would evaluate such a thing by recalling past experiences, asking friends and family, or consulting popular media. Yet with the rise of the internet and online forums, a new method of assessing reasonability of one's behavior has become commonplace; posting online a description of one's behavior so that other individuals may decide if their behavior is acceptable or not.

This is the exact purpose of the subreddit r/AmITheAsshole, a subreddit where Reddit users can post descriptions of real world scenarios where they seek to understand if they were the asshole in a given situation. In addition, they often mention their age and gender, along with the age and gender of other people involved in the situation. After the original poster (OP) makes a post, other users post comments on the post, where they begin their message with one of four rulings: You're the asshole (YTA), Not the asshole (NTA), Everyone Sucks Here (ESH) and No assholes here (NAH). After they post this label, they then explain their reasoning for their ruling, and other users can like (upvote) or dislike (downvote) other commenters. If a post receives enough engagement (e.g. comments, upvotes, downvotes) then the highest upvoted response's judgement will be assigned as the judgement of a post.

This subreddit thus allows for novel analysis of how online communities judge behavior, and can give insight into how a wide swath of people judge actions that could ordinarily only be assessed through costly surveys. In particular, if one could develop a model that could predict the judgement of a post from purely text post data, then one could analyze what features of a post lead to a particular judgement. For example, if we found that, controlling for all potential confounders, men were more likely to receive YTA ratings than women, than this would suggest that not only does the content of the post affect judgements, but the actual identity of the poster themselves. Fitting such a model will be the primary goal of our analysis in this project.

2 Prior Work

There have been a few other studies in this area of research that are relevant to our project. For example, in Botzer et al. [1], researchers used user comments to predict the label of a post with 90% accuracy. They did this by training a model to predict the label of comments, which could thus be leveraged to rule judgement on posts, since this is exactly how labels are determined on r/AmItheAsshole. This kind of analysis, however, does not allow us to understand exactly which aspects of a post lead to different judgement, and additionally, does not allow for analysis of posts for which no comments are present, which limits the scope of their work.

Preprint. Under review.

Similarly, in Haworth et al. [6] researchers achieve an accuracy of 61% using just text features of the post and 77% using metadata from the post using BERT. While this work moves closer to just relying on the posts, it achieves rather poor performance using just text data, and the metadata they do use does not have very compelling interpretations. For example, they use downvotes (dislikes) on a post as a part of their metadata, which is correlated with the ruling YTA, since people tend to get upset at users they see as morally reprehensible.

Finally, in [3], they perform a similar analysis to that of [6], where they user linguistic features to predict post and comments using BERT. They achieve a similar accuracy, and attempt to analyze the implications of their analysis using GrASP, a framework for interpreting why a transformer model like BERT outputs the results it does. While they do come to a few conclusions about how their model makes decisions, the lack of interpretability of a model like BERT makes it difficult to come to any compelling conclusions about why the model deems some posts YTA and other as NTA. Our analysis hopes to improve on all of the above prior work by on building a more interpretative model that achieves at least a similar accuracy, if not better.

3 Data Description

Our data comes from a data dump of publicly available Reddit posts from r/AmItheAsshole during the years 2005-2023, which is about 2 million posts. At the current stage of our analysis, we have only worked with data from 2023, which includes 371,849 posts. After reading in these posts, we then filtered only to look at posts that are labeled as either "YTA, "NTA", "ESH", or "NAH"; these labels are determined by the ruling of the top comment on a post, where the top comment is determined by which comment has the most overall likes (or upvotes). We recoded NAH and ESH to YTA and NTA to make this an easier binary classification problem, since we expect ESH/YTA (NTA/NAH) posts to be similar. This labeling system was developed by the moderators of r/AmItheAsshole, as they have determined this best represents the overall label of the post; we will also use this label, given that we trust the moderators of the subreddit as experts in this case.

After filtering for labeled posts, we were left with about 74,658 labeled posts that we can use for model fitting. For about 3,000 of these posts, the body text was deleted at the time of scraping the data, so that while we have information on the title and the existence of the post, the actual text of the post was removed; except for our topic analysis, we kept these posts present in the data, and used their title instead of the body text for training. These 74,658 labeled posts from 2023 are the data we will work with for the rest of this report. There is a significant class imbalance present in our data (78% NTA, 22% YTA) so we will often need to rebalance our data when training in order to assess accuracy.

4 Problem Statement

For our project, we have two primary aims of our analysis. For the first aim, we will create an interpretable model that assigns labels to posts (YTA or NTA) using exclusively the text of the post itself. We then will interpret this model's output, with the goal of understanding what aspects of a post lead to it being labeled as YTA or NTA. For this purpose, we have used assessed BERT and decision trees to label posts.

For our second aim, we will perform topic classification of the posts, so that we may better understand what kinds of topics people tend to post about on these sites. This will be interesting in its own right, but we also expect that this will be a useful predictor when trying to predict the label of the post, as we might expect that different kinds of topics/behaviors are labeled differently by users on average. For this analysis, we have used BERTopic, a flexible model for unsupervised topic classification that we will discuss in 5.1.

5 Mathematical/Technical Background

5.1 Topic Classification using BERTopic

In order to classify topics, we used a flexible model called BERTopic [4] that allows for us to perform unsupervised or semisupervised topic classification from text. As we can see from 1, clas-

sification for this model consists of 6 main steps. First, we take a set of documents and create vector embeddings of our data using any common embedding method (e.g. word2vec). Then, we apply dimensionality reduction to our embeddings (in order to reduce the curse of dimensionality), and apply a clustering algorithm to the output. This gives us our clusters (or topics) of our documents; next, we need to develop informative labels for these topics. We then tokenize the documents in each cluster, where we now treat each cluster as its own document, and then we apply a modified form of TF-IDF (c-TF-IDF) where instead of calculating TF-IDF on our original documents, we treat each topic cluster as a document, so that we can understand which tokens (in our case, words) best represent each cluster. We then take the top n tokens (as weighted by TF-IDF) to be the label for our unsupervised clusters, so that after these five steps, we have fully clustered our documents into topics, with each topic's label representation being derived from the words/phrases most unique to that topic. Then, we can optionally fine tune these label representations by passing our topic label into a representation model, such as a LLM, that could take in a group of words and summarize them into a more distinctive, fine-tuned topic label.

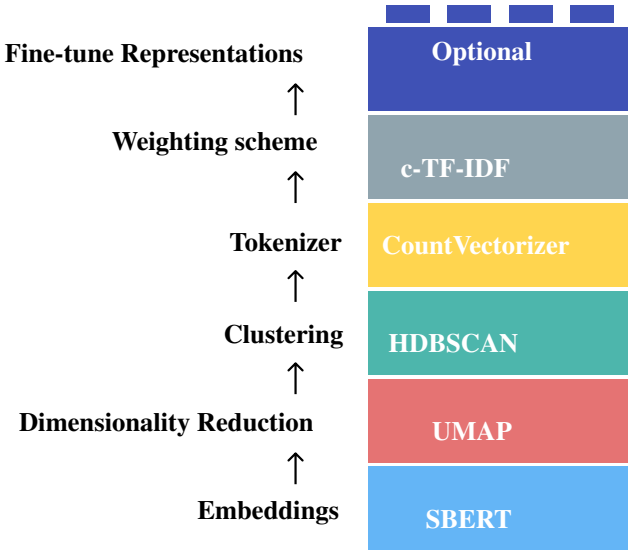

Figure 1: Visualization of BERTopic's Modular Pipeline [5]

In our case, our documents represent the body text for each post. We provide our choices for these models in 1, and were largely chosen following recommended best practices from the author [5].

Table 1: Models selected for BERTopic

Model	Type of Model
Embedding Model	all-MiniLM-L6-v2
Dimensionality Model	UMAP
Clustering Model	HDBSCAN
Vectorizer Model	CountVectorizer
Weighting Scheme	c-TF-IDF
Representation Model	KeyBERTInspired, Maximal Marginal Relevance

5.2 Classification Model: LightGBM

We used one type of implementation of the Gradient Boosting Decision Tree (GBDT) algorithm called LightGBM[7] to classify the labels YTA (You're the Asshole) and NTA (Not the Asshole). LightGBM is a framework of GBDT that optimizes the speed and performance. GBDT is an algorithm that combines three techniques in machine learning: boosting, decision trees, and gradient descent. Boosting is a method where, after building a decision tree with the given data, subsequent trees focus on the data points that were not classified correctly in previous trees. By repeating this step, the model's accuracy improves. Furthermore, in training the decision trees, gradient descent

is used to minimize the objective function. This is done by calculating the gradient of the objective function at each boosting stage and using it to refine the model iteratively until the error is minimized.

LightGBM is computationally faster than other frameworks of GBDT because it uses leaf-wise tree growth. This means that normal decision tree algorithms calculate the tree level-wise, evaluating all the splits at one level first, then moving to the next level. However, LightGBM focuses on splitting the leaves that result in the greatest reduction of loss, optimizing tree growth for improved performance. Another feature of the framework, exclusive feature bundling, bundles covariates that are similar, which also decreases computational cost. Additionally, LightGBM employs histogram-based decision tree learning and Gradient-based One-Side Sampling (GOSS) for better speed and scalability.

We also used SHAP (SHapley Additive exPlanations)[9] to interpret the model. This applies the idea of Shapley value from game theory to quantify the contribution of each feature to the model's predictions, identifying how each feature influences the output and in which direction.

5.3 BERT

To build our first large language classification model, we apply the Bidirectional Encoder Representations from Transformers (BERT) [2] implemented by and distributed on Hugging Face (`https://huggingface.co/`) to our classification task. Specifically, we choose the `bert-base-uncased` model from the BERT family. It is the base model that treats the lower and upper case of any word identically. Hugging Face provides convenient APIs for us to access and use state-of-the-art pre-trained models available from the Hugging Face hub.

A detailed explanation of the structure and the pre-training process of BERT can be found in the original paper [2] or BERT's model card on Hugging Face (`https://huggingface.co/google-bert/bert-base-uncased`), so we will not reiterate it here. Instead, we report and explain critical design choices and the hyperparameters used for data preprocessing, tokenization, and model fine-tuning.

An important preliminary step is to rebalance the dataset. After some basic filtering (e.g., drop observations with ill-formed or missing paragraphs or labels), we found our dataset contains about $58,000$ negative observations and $16,000$ positive observations. Some prior work [6, 3] have already mentioned that this significant class imbalance phenomenon could hurt the fine-tuning procedure. We also verify this after some small fine-tuning experiments. Thus, we subsample negative observations so that the final dataset contains the same numbers of positive and negative observations. In other words, the dataset contains about $32,000$ observations.

We designed two methods to create the textual input to the BERT. The basic method is to combine the title and the main body of a post into a single paragraph as the textual input. The advanced method adds the gender, age, and sentiment scores from other parts of our work besides the title and the main body to the textual input. We wished that the extra features could provide social and emotional information about the post that help improve BERT's prediction. The textual inputs of a random observation with two different methods look as follows:

- Advance: `gender: Female / age: 40 / topic: parents upset stay later / title anger score: 0.0032 / title disgust score: 0.0619 / title fear score: 0.0026 / title joy score: 0.0058 / title others score: 0.9197 / title sadness score: 0.0017 / title surprise score: 0.005 / text anger score: 0.0051 / text disgust score: 0.9394 / text fear score: 0.0033 / text joy score: 0.002 / text others score: 0.0413 / text sadness score: 0.0071 / text surprise score: 0.0018 / title: AITA for not wanted babysit my niece for the entire summer? / text: I am a SAHM (40) with 5 kids (also the oldest of my siblings). Last night my sister (37) told (but acted like she was asking in her entitled way) me I was going to babysit her daughter (7) all summer because she didn't look it to any summer camps/programs and didn't want to pay for them when I am home anyway..."`

Next, for each paragraph, we lowercase it, strip its multiple white spaces, and drop it if it becomes empty in the end.

For tokenization, we use the pre-trained tokenizer for `bert-base-uncased`. We pad or truncate from the end of a tokenized paragraph to make its length 512, which is the maximum possible length of a tokenized paragraph that `bert-base-uncased` can process. We padded or truncated from the end because the reader tends to read the beginning of the post more carefully and gradually lose attention or interest when approaching to the end of the post.

For the fine-tuning procedure, we list all fine-tuning hyperparameters in Table 2. The seed hyperparameter is the random seed applied to subsampling, train-val-split, and Hugging Face's trainer object. The weight decay hyperparameter is applied to Adam optimizer with weight decay fix (`AdamW` in PyTorch) [8]. The evaluation step being 200 means for every 200 iteration or batch, we evaluate the current status of our model (i.e., model checkpoint) and save it locally. At the end of the fine-tuning, we load back the model checkpoint with the best F1 score. We will show the performance of the best model checkpoint under this fine-tuning procedure in section 6.3.

For the hyperparameter tuning, we tried several different combinations of hyperparameters, like learning rate, weight decay, and dropout rate. However, the F-1 scores differ within 1%. Moreover, BERT is known for being robust to hyperparameter tuning. Therefore, we did not really tune the hyperparameters and stick to the hyperparameter values shown in Table 2 that were recommended by some other work.

Table 2: Hyperparameters of the fine-tuning procedure.

Hyperparameter name	Hyperparameter value
Classifier Dropout	0.15
Learning rate	5.00E-05
Learning rate scheduler	Cosine with warmup
Batch size	16
Gradient accumulation step	4
Weight decay	2.00E-03
Epoch	5
Evaluation step	200
Evaluation metric	F1 score
Seed	547

6 Results

6.1 BERTopic

After running BERtopic on our data, with our current model configuration and hyperparameters (the only hyperparameter currently of note is min topic size, which we set to be 100), we received approximately 55 genuine clusters, and 1 cluster for points that were not able to be effectively clustered. In Table 3 Upon initial inspection, BERTopic appeared to be picking up on genuine topics that exist in the data. For example, our third most popular topic with 2,662 posts had the label " [wedding, bridesmaids, engagement, party, bach...]", which upon inspection, appears to represent posts where people are concerned about wedding planning, a common source of stress where oftentimes moral dilemmas arise. Note that topic 0 in Table 3 corresponds to the unclustered points, but even within this cluster, we can see that parents is the most identifiable word, indicating posts involving families (and parents) are quite common on r/AmItheAsshole.

Table 3: Summary information for the first 10 topics

Topic ID	**Count**	**Topic Representation**
0	30417	[parents, house, stay, later, upset, make, say...]
1	10453	[friendship, friend group, bf, upset, hurt...]
2	2662	[wedding, bridesmaids, engagement, party, bachelor...]
3	2497	[custody, pregnancy, upset, mom, relationship,...]

Topic ID	Count	Topic Representation
4	2413	[savings, financial, afford, mum, college, sibling...]
5	2226	[barking, animals, leash, neighbor, walk, stay...]
6	2157	[meals, cook, foods, dish, chicken, restaurant...]
7	1571	[cleaning, chores, laundry, dishes, roommates,...]
8	1517	[holidays, thanksgiving, family, celebrate...]
9	1276	[pets, kitten, litter box, shelter, house...]
10	1249	[roommate, pay rent, lease, house, moving...]

After getting these categories, we calculated the proportion of each label (YTA, NTA) in each category to assess category heterogeneity. Overall, we found that the proportion of YTA from .07 to .3, depending on the category, indicating that the category of the post is certainly correlated with the label of the post, justifying our decision to use it in the forthcoming models. Interestingly, the categories where there were lower proportions of YTA ratings were often topics one might expect to be controversial (e.g. ['stepdad_divorce'], ['trans_lgtbq community'], ['mum_mental health']), which may indicate that posters in these categories were more likely to be the victim of some harm rather than the perpetrator. On the other hand, categories with high "YTA" ratings were a mix between serious topics (['affair_best friend'], ['family_brother')] and less serious topics (e.g. ['left_lane traffic'], ['swimming pool_children'], ['xbox_laptop']). This might indicate that categories with high YTA ratings are situations which are really exceptionally morally frowned upon, as well as those that might constitute more minor situations. This seems to make sense, as users (OPs) might be more likely to post about situations in which they may be the asshole in less serious situations, since the moral consequences wouldn't be as clear or as high.

In addition to the above plots, BERTopic is able to produce hierarchical topics, such that each topic in the model belongs to a "super-topic" that is a cluster of related topics. This was one of the more interesting parts of our analysis, as it allows us to answer the question "What, generally, do people post about on r/AmItheAsshole" with varying degrees of specificity. From Figure 6.1, we can see that there are roughly 5 major topics that people tend to post about on the subreddit: specifically, the green topic appears to be about traffic, the red topic appears to be about household concerns/roommate problems, the teal category (the largest category) seems to deal with familial relationships, friendships and work, the purple category has to deal with weddings, and the yellow category has to do with money and gifts. The number next to the topic label indicates its ranking in the number of posts compared to other topics. Additionally, by following the tree downward within categories, we can see lots of interesting subcategories; for example, in the red category of roommate/household issues, we can see there is a subcategory corresponding to animals. Interestingly, [wedding_bridesmaids] is not clustered with [dress wedding_bridesmaids], indicating that we should take some of this hierarchical clustering with skepticism. However, the overall takeaway does seem to be relatively robust; people tend to post about stressful situations, confrontations with others, financial concerns, and household issues, with particular emphasis on animals and weddings.

6.2 LightGBM

6.2.1 Data Used for the Model

We used lightGBM to make a model for predicting YTA and NTA labels to interpret the model better. Using the data from section 3, we built four categories of features as shown in Table4 with various methods. For the first category linguistic, as in previous studies, the number of words and the length of sentences, as well as the number of demonyms, were extracted from the titles and body text of the posts. We used dictionaries of words for demonyms[1], stop words[2], and profanity words[3]. The gender and age features were also extracted from the posts. This is based on the fact that, in the r/AmITheAsshole subreddit, it is customary for posters to represent the gender and age of the people involved using symbols like "My bf (25M) and I (25F) are ..." Initially, we applied simple rule-based methods, achieving an initial determination rate of 58.8% for gender and 46.2% for age. For those posts where gender and age could not be initially determined, we employed the

[1] https://github.com/porimol/countryinfo

[2] https://github.com/nltk/nltk

[3] https://github.com/zacanger/profane-words

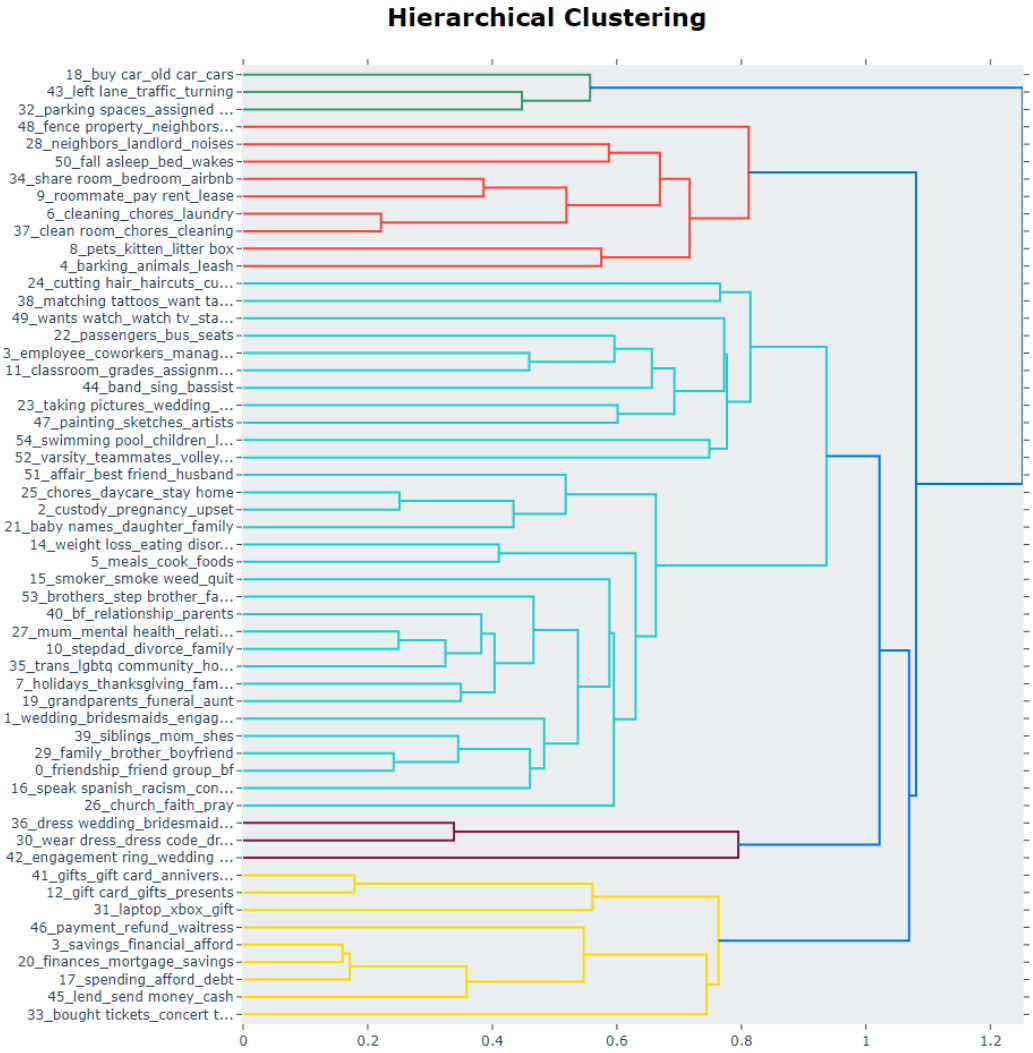

Figure 2: Percentage of "YTA" posts in top 30 categories by size

Llama 3[4] `8b-chat-hf` version. Specifically, we gave the model a prompt such as "Extract mentions of the author's gender and age from the title and body of this post." As a result, we were able to determine the gender for 64.9% of the posts and the age for 53.4%. For the sentiment features, we employed pysentimiento[5], which is a transformer-based library for various NLP tasks. It outputs the probability of each sentiment type: anger, disgust, fear, joy, sadness, surprise, and others. Topic features are from the BERTopic analysis we have explained in section 5.1. The model was also trained on data that contained 50% YTA and 50% NTA, and test data was also rebalanced too.

6.2.2 Results and Findings

The performance metrics of the model are shown in Table 5. Combining features resulted in improved model performance. For instance, using only topic features yielded an F1-score of 0.51, whereas combining all features increased the F1-score to 0.57. However, even with all the features, the model did not perform well, as we have seen with BERT in Section 6.3, especially considering that a baseline model predicting all posts as YTA would still achieve an accuracy of 50%. This suggests that there are likely other elements influencing people's judgments that are not captured by these features.

[4]https://github.com/meta-llama/llama3

[5]https://github.com/pysentimiento/pysentimiento

Table 4: Feature names and descriptions

Category	Feature Name	Feature Description
Linguistic	title_uppercase_count	Num. of capitalizations in title
Linguistic	title_word_count	Num. of words in title
Linguistic	title_profanity_count	Num. of profane words in title
Linguistic	avg_word_length	Avg length of words in post
Linguistic	stop_word_count	Num. of stopwords in post
Linguistic	numerics_count	Num. of numbers in post
Linguistic	uppercase_words_count	Num. of capitalizations in post
Linguistic	sentence_count	Num. of sentences in post
Linguistic	avg_sentence_length	Avg num. of words per sentences
Linguistic	profanity_count	Num. of instances of profanity in post
Demography	demonyms_word_count	Num. of demonym words in post
Demography	demonyms_unique_count	Num. of unique demonym words in post
Demography	gender	Gender of post author
Demography	age	Age of post author
Sentiment	title_sentiment	Probability of each sentiment in title
Sentiment	sentiment	Probability of each sentiment in post
Topic	topic	Topic ID of post from Table 3
Topic	representative_words	Representative words from Table 3

Feature Set	**Accuracy**	**Precision**	**Recall**	**F1-score**
Linguistic Features	0.54	0.52	0.52	0.52
Demography Features	0.56	0.56	0.56	0.55
Sentiment Features	0.52	0.52	0.52	0.52
Topic Features	0.54	0.54	0.53	0.51
All Features	0.57	0.57	0.57	0.57

Table 5: Performance metrics of the LightGBM model for different feature sets

Figure 3 illustrates the SHAP values for each feature. Higher SHAP values indicate that the feature influences the model to predict the post as YTA, while lower values indicate an influence toward predicting NTA. Features with SHAP values centered around 0 have minimal influence on the model. The color gradient represents the feature's value. For instance, stop_word_count has red points on the left side of the graph and blue points on the right, signifying that a higher count of stop words leads the model to predict the post as NTA, while a lower count suggests a YTA prediction.

The figure clearly shows that as the emotion conveyed by the title becomes sadder, the higher the probability of the post being judged as NTA. In other words, titles expressing sadness may influence readers to sympathize with the author. Interestingly, the top sentiment features by SHAP value are ranked higher for the title than for the post text itself. This suggests that people may be biased by the title when they first see a post. On the contrary, posts expressing fear, joy, or surprise are more likely to be judged as YTA. Also, posts by male authors tend to receive more YTA labels, and posts authored by older individuals are also more likely to be predicted as YTA. These findings suggest that the author's emotion and role significantly influences how people perceive the post's morality, which affects whether it's labeled YTA or NTA. The author's role and their relationships with others, as described in the post, could be valuable features to include in future work. Lastly, there are also some findings that are difficult to interpret intuitively. For example, the reason why stop words or average word length had such a impact on the model's predictions, is what we need to investigate further.

6.3 BERT

With the data preprocessing, tokenization, and fine-tuning procedure described in section 5.3, we show the performance metrics of our fine-tuned BERT model on the validation set with two different textual inputs in Table 6.

The basic BERT model achieved an F1 score of 0.6691, and the advance BERT model achieved an F1 score of 0.6520. It is surprising to us that providing the extra features does not help improve the

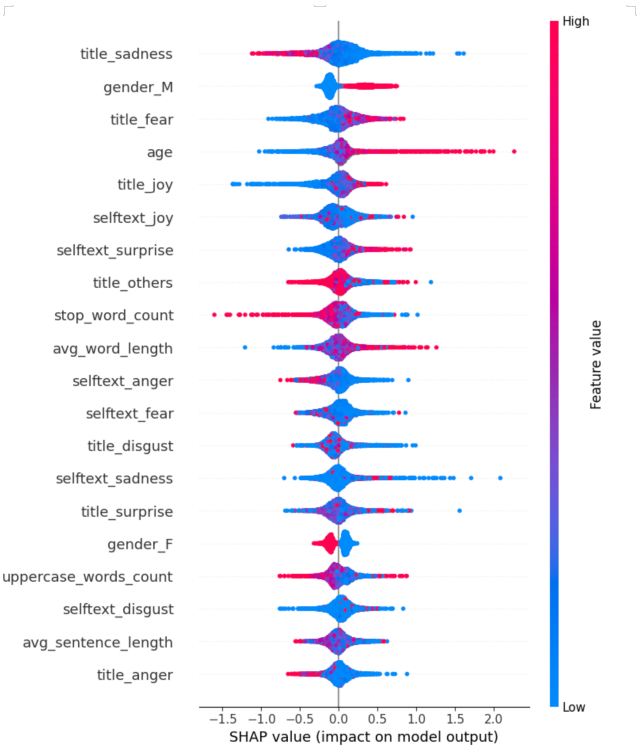

Figure 3: SHAP value for top 20 features: `gender` is encoded to one-hot vector (F for female, M for male, U for unknown). Features with the prefix `selftext_` represent sentiment expressed in the post text.

Table 6: Performance metrics of the best BERT model checkpoint on the validation set.

	Basic	Advance
Accuracy	0.5902	0.5960
F1-Score	0.6691	0.6520
Precision	0.5562	0.5670
Recall	0.8395	0.7671
ROC AUC	0.5934	0.5982
Specificity	0.3474	0.4293

prediction significantly. This supports the power of BERT to capture diverse information from pure text. However, both models significantly outperform the best models in [3] (0.61 by BERT) and [6] (0.61 by Random Forest). Note that we consider the model on linguistic features only in [6] because our BERT was fine-tuned on the textual paragraphs only and not on any non-textual features in [6]. Therefore, taking into account that our dataset is larger and more recent than those in [3, 6] and this might benefit us, we claim that our BERT model performs at least as good as the best models in these prior works.

One interesting observation on our BERT model is that our model performs better on positive observations (i.e., YTA) than on negative observations (i.e., NTA) as recall is higher than specificity. One possible explanation is that there exist some strong keywords in the title or the main body of YTA posts such that our BERT model can easily capture these keywords. It is worth exploring this observation further in the future.

Another interesting observation is that the advance model has a higher specificity than the basic model. This means that the advance model makes fewer type-I errors or false positives than the basic model. It might be because adding the sentiment score features allows the model to capture the negative sentiment more accurately. Given that calling someone an asshole is a relatively serious accusation, it might be wiser to use the advance model.

7 Difficulties

The challenge in this task lies in creating a model that is interpretable and helps us understand how people judge the morality of a post. Transformer models like BERT often outperform decision tree-based models, but interpreting BERT's attention mechanisms is more complex than observing SHAP values. It is challenging to identify the features that influence people's moral judgments of posts and to accurately quantify and extract those features for tree-based models. Without features that reveal the content and tone of the post, the model will lack crucial information for accurate labeling.

Furthermore, factors that drive people to vote YTA or NTA can be influenced by two main aspects: the actual content or tone of the post. Although it's difficult to measure the logical reasoning behind how people perceive posts, we attempted to capture the content and tone through topic analysis and sentiment analysis. For our topic analysis, our primary difficulties come from the fact that this kind of modeling is entirely unsupervised.

8 Future Work and Conclusions

8.1 BERTopic

Overall, we found that people tend to post about stressful situations (like weddings), roommate disagreements, interpersonal relationships, financial difficulties, and annoyances like driving. In our future work, we hope to fine tune the topic model so that we have more confidence in our topic labels. This will largely involve some form of sensitivity analysis, where we adjust not only the hyperparameters of each modularized model in TopicBERT but the model class itself. For example, we could use ChatGPT to create our topic labels to see if we get different results compared to other representation models.

8.2 Modeling judgements

In our work, we developed two models to classify text into YTA/NTA categories. The BERT model achieved higher scores on several accuracy metrics, including an F1-Score that was 10% better than prior works. Regarding model interpretation, we were able to interpret only the LightGBM model, yielding somewhat feasible results. SHAP values revealed that the demographics of the author, such as gender and age, are associated with moral judgment. Additionally, the sentiment of the title was more strongly associated with the judgment than the text itself. Future work could involve interpreting BERT models further to gain a deeper understanding of moral judgment and exploring other models, such as Longformer, to potentially increase accuracy. From this work with LightGBM, we determined that including the demographics of the author increases model accuracy. To further enhance the LightGBM model, we could incorporate features such as the role of the post's author, which might reveal additional characteristics of the author and their relationship to the people they are having issues with.

References

[1] Nicholas Botzer, Shawn Gu, and Tim Weninger. Analysis of moral judgment on reddit. *IEEE Transactions on Computational Social Systems*, 2022.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Ion Stagkos Efstathiadis, Guilherme Paulino-Passos, and Francesca Toni. Explainable patterns for distinction and prediction of moral judgement on reddit. 2022.

[4] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure, 2022.

[5] Maarten P. Grootendorst. The algorithm¶. URL `https://maartengr.github.io/BERTopic/algorithm/algorithm.html`.

[6] Ethan Haworth, Ted Grover, Justin Langston, Ankush Patel, Joseph West, and Alex C. Williams. 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, May 2021. doi: 10.1609/icwsm.v15i1.18133. URL https://ojs.aaai.org/index.php/ICWSM/article/view/18133.

[7] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3149–3157, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[9] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874, 2017. URL http://arxiv.org/abs/1705.07874.