Are You A Bad Person? - Diving Deep Into Reddit Morals

Zain Alden Jaffal **Teodora Reu**

Abstract

Judging a person's morality in the context of a story depicted by them can be a very subjective matter. A listener when making a decision or a judgment will not only take into consideration that person's story, but also past experiences, biases, and previously received an education, etc., i.e. their opinion will be subjective and maybe different from others' opinion. This project proposes various benchmark machine learning models and attempts a classification task on stories told by subjects on AITA (Am I The A**hole?) subreddit to tell whether the subject is a bad person or not. Furthermore, we propose a novel hierarchic integrative model featuring BERT and benchmark models based on a combination of Doc2Vec and classifiers such as Naive Bayes and Multi-Layer Perceptron which attempt classification on the concatenation of the hidden space representations of both posts and comments. This work shows, that sentiment classification on posts can benefit from further analysis of the comments, i.e. of the *popular* opinion. By evaluating the integrative novel model with metrics such as F1, we will notice a rise in the overall accuracy, with percentages that exceed the benchmark ones with 10%.

1 Background on Sentiment Analysis

Sentiment analysis represents a very popular subject in research nowadays. In this section, we will introduce efforts and state-of-the-art models that attempt the classification of text into sentiments.

Birjali et al. (2021) introduces traditional approaches such as taking a vector embedding of the text using methods like BagofWords model, GloVe, Word2Vec or **Doc2Vec** (Le and Mikolov, 2014) on which a combination of learning algorithms such as Naive Base or SVMs (Joachims, 1998) are used to classify the text; and secondly, Deep learning approaches which show promising results in comparison with the models already mentioned. Long Short Term Memory Neural Networks (**LSTMs**) (Hochreiter and Schmidhuber, 1997) have massively improved RNNs to address the problem of vanishing gradients.

Fine-tuned **BERT** can also perform very well on sentiment analysis tasks. In this paper (Sun et al., 2019), authors experiment with different fine-tuning methods on 3 different datasets (IMDb, Yelp P., Yelp F.). Their models obtained leading test error rates like 4.37%, outperforming all other models like LSTMs and CNNs. Additionally transformers like SMART-RoBERTa Large (Jiang et al., 2019) obtained 97.5% accuracy for STT-2 binary classification sentiment analysis task.

2 Related Work

On the task of studying **AITA**, Haworth et al. (2021) classified the posts based on the post and user's metadata using various traditional machine learning methods like Random Forest classifiers, Logistic Regression, Support Vector Machines, k-Nearest Neighbors, and Naive Bayes classifiers.

For the sentiment analysis on comments Botzer et al. (2022), the authors build a model for Empathy/Hate Classification based on the assumption that a negative comment (not-emphatic) would appear only in the context of a YTA post, and reversely a positive emphatic comment would only appear in the context of an NTA post.

Tag	Meaning	Sentiment	Count
YTA	You're the A-hole	Negative	372,850
NTA	Not the A-hole	Positive	717,006
ESH	Everyone Sucks here	Negative	79,059
NAH	No A-holes here	Positive	91,903
INFO	Not Enough Info	Neutral	-

Table 1: AITA labels, commentors pick one of the specified labels. Comment counts are taken from Botzer et al. (2022) work on the subreddit January 1, 2017, and August 31, 2019

	Posts		Comments	
Tag	YTA	NTA	YTA	NTA
Counts	4474	10868	7284	17702
Ratio	0.29	0.71	0.29	0.71

Table 2: AITA labels ratio and count based on collected posts and top three comments for each post

Level of agreement	NTA agreement	YTA agreement
3.0	0.94	0.87
2.0	0.05	0.08
1.0	0.01	0.04
0.0	0	0.01

Table 3: Level of agreement between posts and comments broken down by label

3 Methods

This section will provide information on how the `r/AITA` dataset was obtained, preprocessing methods applied to the data, and descriptions for all proposed models

3.1 Dataset

The dataset was derived from `r/AITA`. In the forum, each post is classified into 4 tags listed in Table 1. Given the majority of the posts are classified into YTA and NTA we ignored the rest of the labels making our task a binary classification problem.

Our dataset contains both posts and comments obtained in two stages leveraging *pushshift.io* and Reddit API. First, we pulled posts, together with their ID, TITLE, BODY, and LABEL (either YTA or NTA). The post's query doesn't return comments so we used the REDDIT API to pull the top-level comments for each post, sorted by their upvotes. We discarded any post from moderators, bots, and any post that didn't contain the labels YTA and NTA. This way we ensured that the comments were classified.

This process has proved to be a very time-consuming reason which is why we decided to extract 3 comments per post. For each comment, we saved its ID, BODY, and LABEL.

We note the following characteristics of the dataset:

1. Most of the time commentators agree with the label of the post, as displayed in table 3.

2. Generally, the comments follow this structure:

   > *[YTA/NTA] I find it hard to believe that someone who's in college and can speak three languages fluently can't handle themselves to a certain degree . . .*

   This is important for classifying the comments by automatically tagging them using regular expressions.

3. The dataset contains predominately negative samples (NTA) Table 2. By randomly classifying the dataset, we can achieve an accuracy of 70%. To counteract this we undersampled our data.

4. From figures 3a and 3b, we can see that the average length of a post is around 329 and 43 words respectively additionally 15% of the posts are longer than 512 words which is the maximum number of tokens for BERT.

We collected posts and comments between the period 01/Jan/2019 - 01/Jan/2020 to make sure that all the posts we collected were tagged and that there were no ongoing comments or post updates. We collected 10k posts and 30k comments. We split them into three datasets one for posts one for comments and a dataset containing both a post and the top 3 comments. The latter dataset consists of only 8k posts. After performing undersampling we end up with 9k posts, 16k comments and 4.8k merged posts and comments.

In our experiments, we split our data into three types. The majority of the data is used for training. During training, we have a validation dataset running after each training epoch. Finally, all models are evaluated on the testing dataset.

3.2 Benchmark Models

We used two methods as baselines for our project. The first is a traditional machine learning approach using Naive Bayes or a Multi-Layer Perceptron (MLP) to classify the post from a latent representation. We learned an embedding vector using **Doc2Vec** which learns the latent representations for each post in an unsupervised manner. For preprocessing we have lowered all words, removed links, and applied usual tokenization on posts. We imported the model from *gensim* (Lau and Baldwin, 2016), and initialize it with $vector\_size = 300$, $negative = 5$, $hs = 0$, $min\_count = 2$, and $sample = 0$.

The second approach used an **LSTM** model with 2 hidden layers and a vector size of 500. The model takes the post-word-by-word learning an embedding representation and inferring a label after going through all the words in the post.

Both models were trained using the posts dataset without undersampling. For the second approach the model was trained for 10 epochs with a constant learning rate of $1e-4$ and batch size of 4.

3.3 Proposed Model

Our proposed model follows a hierarchical architecture and consists of two steps. The first step is to hierarchically classify the post and top 3 comments using two different BERT models. The next step is a head layer taking the hidden representations from the `[CLS]` token from both models and classifying the post into either YTA or NTA.

The assumption is the model will understand multiple perspectives. First by looking at the story and interpreting from the context if the narrator is a bad person or not. Secondly, the commentators offer a quick and short explanation of why they judged a person in a certain way.

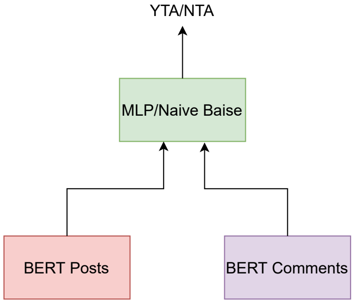


Figure 1: Proposed model structure

3.3.1 BERT on Comments

We trained the model on a version of the comments where the commentator's judgment is hidden. The model should infer the judgment from the text. We used the basic BERT model with 128 tokens covering 99% of the collected comments lengths. Followed by a `Dropout` layer then a linear layer mapping BERT's latent vector into a vector of length 2 where $\texttt{YTA} = [0, 1]$ and $\texttt{NTA} = [1, 0]$.

The model is trained on the comments data set using a training, validation, and testing split of 13111, 1456, and 1776 comments respectively. We note that all datasets have been undersampled. For preprocessing, we followed the same methodology as the one described in the previous section. The comments were tokenized using a pre-trained BERT tokenized.

The model is trained for 3 epochs with a linear learning rate starting with 100 warm-up steps and an initial learning rate of $2e-05$ using Adam optimizer training is similar to Botzer et al. (2022). The loss function used is Binary Cross-Entropy Loss (BCE).

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{N}^{i=0} y_i \cdot log(\hat{y}_i) \\ + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

3.3.2 BERT on Post

BERT on post shares a similar structure to BERT on comments. Instead of classifying on comments the model takes the entire story and infers the tagged label of the post. This BERT model was instantiated with 512 tokens - the maximum token length used by BERT. We note that majority of the posts ar

The model had the same preprocessing, tokenization and training steps as Bert Comments but with 8053, 894, 756 split for training, validation and testing datasets.

3.3.3 Joint Bert

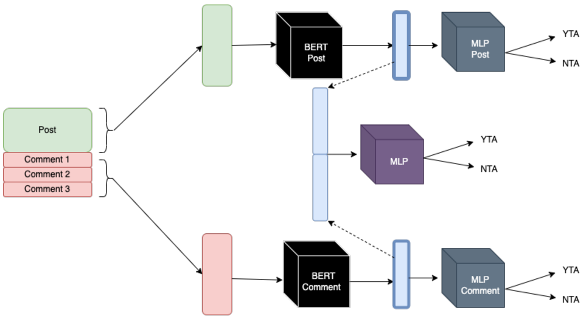


Figure 2: Hierarchic Joint Bert

After training both Bert Comments and Bert Posts. We integrated the hidden features of both the post and it's respective comments. We introduced a head layer taking the hidden representations from the `[CLS]` token and classifies the post into either YTA and NTA.

The total embedding vector x is the concatenation of the latent post vector h_{post} and the average of comments embeddings $h_{comments}$.

$$h_{comments} = \frac{1}{N}\sum_{i=1}^{N=3} c_i \tag{1}$$

$$x = h_{post} || h_{comments} \tag{2}$$

Following this we have a function to estimate the class labels based on the total embedding vector x.

$$\{\text{YTA}, \text{NTA}\} = \text{head}(x) \tag{3}$$

In our model, we experiment with head(x) as a Naive Bayes model and as a Multi-Layer Perceptron (MLP) with two layers with a hidden dimension 500 for each layer. To train the head module, we froze both BERT models and only train the head using the merged dataset with undersampling. For MLP we use the default settings for training provided from sklearn with BCE loss function.

Both models were trained on the merged dataset containing a post and top 3 comments with undersampling using 3837 samples for training, 427 for validation, and 550 for testing.

4 Evaluation

In order to evaluate our models we performed classification task on the testing datasets. To access our models we used the following metrics:

- Accuracy, Precision, Recall, F1-score
- Matthews Correlation Coefficient (MCC): shows good results with imbalanced dataset. MCC evaluates the model by looking at all the quadrants of the confusion matrix. MCC returns a high score if model predicts the majority of positive and majority of negative classes correctly. The value of MCC ranges between $[-1, +1]$. Where -1 and +1 indicates a perfect anti-coloration and perfect classification respectively and 0 indicates random classification (Chicco and Jurman, 2020).

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FN) \cdot (TN+FN)}}$$

In the methods section of this paper a detailed description has been given of the evaluation of each model in terms of the dataset splitting. We have chosen positive labels to represent the YTA, and negative labels to represent the NTA.

Metric	Accuracy	Precision	Recall	F1-score	MCC
Doc2Vec+NB	0.83	0.21	0.01	0.02	0.01
LSTM	0.51	0.50	0.52	0.51	0.01
BERT Comments	0.73	0.70	0.78	0.74	0.46
BERT Posts	0.54	0.52	0.79	0.63	0.09
BERT Joint (MLP)	0.70	0.70	0.72	0.70	0.4
BERT Joint (Naive Bayes)	0.56	0.54	0.75	0.63	0.12

Table 4: With blue the benchmark models, with purple the simple models, and with orange the joint BERT models

5 Results Discussion

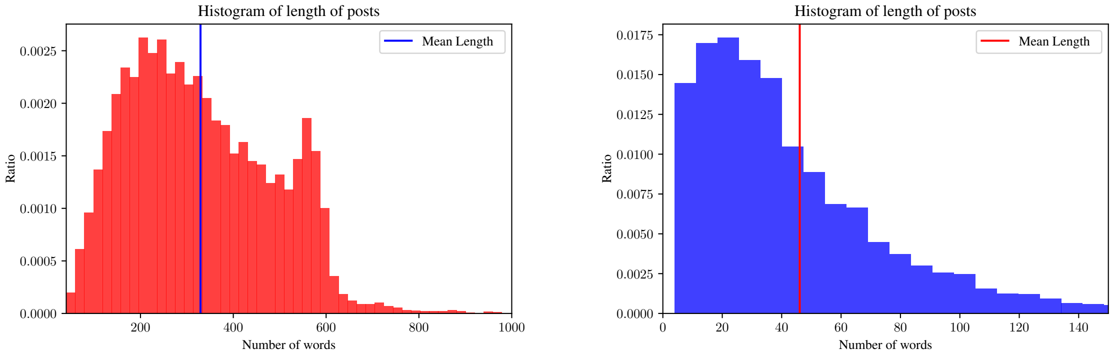
From table 4 we can learn that Benchmark Methods, Doc2Vec+NB, and LSTM, will perform generally worse than BERT models. Even though the Doc2Vec+NB method appears to return a good accuracy i.e. 83% on the post-classification task, it will return poor results for all other metrics classes, e.g. 0.21 Recall and values between 0.01 and 0.02 for the other metrics. The very small MCC metric points out the randomness of the choices of the model. LSTM will perform slightly better than Doc2Vec+NB as expected, with metrics between $0.51 - 0.52$. From the weak results in the case of these benchmark methods, out of which the Doc2Vec+NB is completely non-parametric, we can learn that the classification task on the AITA is indeed a **hard** one.

From the next two rows of the table, we can learn that comments can reveal more sentiment than a post, this being quantified in the metrics obtained. The results for BERT on comments than BERT on posts could be significantly better because comments are much more concise than the parent post. Furthermore, this is likely since we take into account only 512 tokens when we input a post. As we noted in section 3.1 there is still a significant amount of words longer than 512 words.

The last two rows in the table depict the performance of the Joint BERT. As expected the integration of the information coming from both comments and posts increased the value of the overall metrics. Something to note is that the integrative model does less well than the BERT on comments. This can happen because after we obtain the representations on comments we average the obtained representations, instead of concatenating them and applying a Dense Layer on top to reduce dimension. The information may be lost or canceled when averaging, whereas when using Dense Layer, the important sub-features will be preserved.

6 Conclusion

In conclusion, the dataset used is very hard to classify. The stories are long and not uniform and generally use slang language. We demonstrated that the joint model offers better overall performance. The BERT post comment only classifies a portion of the text - given the limited token length of our BERT model- taking the comments into account can help the model understand a better context. For future work we take average embeddings of each



(a) Histogram of word length in posts and mean value

(b) Histogram of comments length in posts and mean value.

Figure 3: Posts and Comments word length distribution over the obtained results from r/AITA

sentence in the post and have a classifier on the resulting latent representation. We can use alterative models for classifying texts. For example, Longformers have been introduced to handle large text or we can use CNNs. Finally, we can apply the model on alternative subreddits and evaluate the general sentiment of the given subreddit.

References

Marouane Birjali, Mohammed Kasri, and Abderrahim Beni-Hssane. 2021. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226:107134.

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

Davide Chicco and Giuseppe Jurman. 2020. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13.

Ethan Haworth, Ted Grover, Justin Langston, Ankush Patel, Joseph West, and Alex C Williams. 2021. Classifying reasonability in retellings of personal events shared on social media: A preliminary case study with/r/amitheasshole. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 1075–1079.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2019. Smart: Robust and efficient fine-tuning for pretrained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer.

Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.