Evaluating interpretability of state-of-the-art NLP models for predicting moral values

Ionut-Laurentiu Constantinescu , Enrico Liscio , Pradeep Murukannaiah
TU Delft

Abstract

Understanding personal values is a crucial aspect that can facilitate the collaboration between AI and humans. Nonetheless, the implementation of collaborative agents in real life greatly depends on the amount of trust that is built in their relationship with people. In order to bridge this gap, more extensive analysis of the explainability of these systems needs to be conducted. We implement LSTM, BERT and FastText, three deep learning models for text classification and compare their interpretability on the task of predicting moral values from opinionated text. The results highlight the different degrees to which the behaviour of the three models can be explained in the context of moral value prediction. Our experiments showed that BERT, current state-of-the-art in natural language processing tasks, achieves the best performance while also providing more interpretable predictions than the other two models.

Keywords: Moral foundations, Moral values, Natural Language Processing, Explainable AI

1 Introduction

Personal values are the abstract motivations that drive our opinions and actions. Understanding personal values can play an essential role in achieving beneficial AI, aimed at creating value-aligned artificial agents that can operate among us. However, estimating personal values is challenging due to their abstract and subjective nature. The recent improvements of Natural Language Processing (NLP) techniques [1], [2] enable the estimation of personal values from opinionated text. However, experiments have been limited in volume and size and often did not study the underlying behaviour that determines models' decisions.

There is a growing body of literature that recognises the importance of explainability in machine learning. In general, it is essential to evaluate the interpretability of any ML model for multiple reasons. One of them is related to the process of building models. Without knowing precisely what the model is doing, improving it for better performance is "a shot in the dark" [3], being only a matter of trial and error. Better understanding can lead to better systems, less bias, and more accountable systems. Furthermore, having transparent algorithms leads to more trust in these systems, which is another crucial factor that currently hinders the inclusion of AI technologies in society. In the end, to use the AI, we need to know exactly what the AI is doing.

However, evaluating the interpretability of a model is not a trivial task, a lot of research around this subject being currently performed. In this specific case, the difficulty comes from the abstraction that the moral values have, as they are high-level social concepts developed by humans that are much harder to understand in terms of exact computer logic. In general, there is no ground truth about the moral values that can be identified in a particular piece of text, this classification being highly relative to the person interpreting it (different people can have different opinions on the same text, and all might be valid and correct views). As a consequence, having no exact definitions for the moral "classes" as you would have for any other classification problem (such as separating images of cats from images of dogs), it is even more critical to understand the reasons behind a specific prediction of a model. In theory, any value can be identified if there is a strong argument supporting the presence of that value in the text.

In this paper, we will focus on evaluating the interpretability of such models, performing a qualitative comparison and evaluation of moral value prediction models based on their interpretability. The main question that we try to answer is:

How interpretable is the prediction of moral values, and to what extent can the behaviour of the different models trained for this classification task can be explained?

Contributions We are going to deliver a well-documented qualitative evaluation of the interpretability for several state-of-the-art NLP models that were used on the task of estimating moral values, such as LSTM, BERT, and FastText. The focus will be on evaluating the interpretability based on visualisation techniques and generally observed behaviour. The results will provide insights into how these models understand and make predictions for moral values, opening the discussion for a more in-depth analysis in the future.

2 Related work

The problem of estimating moral values from text is not novel. Several studies that combine Values, Ethics, and NLP have been performed in recent years. Most of these studies are based on the Moral Foundation Dictionary (MFD) [4], [5], a vocabulary of words associated with a set of moral values. Recently, Hoover et al. [6] also published the Moral Foundations Twitter Corpus, a dataset of around 35,000 tweets annotated according to the Moral Foundation Theory (MFT) foundations [7]. The corpus contains texts from seven different topics regarding critical social issues ranging from the 2016 US presidential elections to the All Lives Matter movement. This dataset has allowed a more in-depth analysis of the performance of NLP models for moral text classification. The current state-of-the-art results in predicting moral values are achieved by [8] which presents a new lexicon entitled *MoralStrength* as an extension to the MFD. The authors applied this new lexicon to the MFTC corpus, training a model for each combination of moral foundation and corpus subdataset.

Although previous studies show to what extent current state-of-the-art models are suitable for text classification for moral values, they do not highlight the degree to which the decisions made by these models can be explained. This is an essential aspect that needs to be considered before deploying such models in real-world applications. Our study aims to fill this research gap by evaluating the interpretability of different models in predicting moral values.

In his paper [9] T.Miller describes interpretability[1] as "the degree to which a human can understand the cause of a decision". He argues that the starting point for explanation in artificial intelligence comes from observing how humans give explanations to each other. While most of the work in explainable artificial intelligence is based only on the "researchers' intuition of what constitutes a 'good' explanation.", his work combines research from philosophy, psychology, and cognitive science in order to present people's social expectations towards the explanation process.

Even though several explainability methods have been proposed in several papers, there is no agreed methodology for performing such an analysis. There is a broad range of methods that try to explain machine learning models' behaviour, and usually, each of them has a different approach to explainability. However, [10] have constructed the XAI question bank, a set of prototypical questions users might ask about AI systems. Furthermore, a recent paper by [11] has investigated how well these questions are answered by current work in NLP and has enhanced the XAI question bank table by including the percentage of studies that attempt to answer these questions, as well as the methods used to tackle these questions. The table can be seen in Figure 1 and will be used as guidance for tackling the interpretability evaluation part.

[1] According to the definition by T.Miller we will use both the terms *interpretable* and *explainable* interchangeably. It is also important to distinguish between the terms *explainability* and *explanation*. We will use *explanation* for explanations of individual predictions.

3 Methodology

The main goal is to evaluate NLP models on the moral value prediction task; hence, we face a multi-label multi-class text classification problem. Formally, this means training a model that given an input sentence X of variable length, predicts an output $y = \{c_1, c_2, \ldots, c_n\}$ containing labels from a predefined set $L = \{l_1, l_2, \ldots, l_m\}$ of moral values. The prediction represents the moral values inferred from the input sentence.

In order to accomplish this task, we need to (1) acquire the necessary data and perform any required pre-processing, (2) implement and train the NLP models on the collected data, and (3) evaluate the models' performance and conduct the interpretability analysis on the best performing configurations.

3.1 Models

Three popular machine learning models have been chosen for evaluation.

LSTM

Long short-term memory (LSTM) [12] is an artificial recurrent neural network architecture commonly used in deep learning. LSTM's no longer provide the best performance in the NLP field but will mainly be used to provide a fair comparison between our results and the results from previous work regarding moral text classification [6].

BERT

Bidirectional Encoder Representations from Transformers (BERT) [1] is a Transformer-based machine learning technique for natural language processing pre-training. BERT's key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. It will be the primary model that we will focus on as it is the model that currently achieves the state-of-the-art results on NLP tasks, including text classification.

FastText

FastText [13] is a model with comparable performance with the state-of-the-art deep learning classifiers, but being many orders of magnitude faster for training and evaluation.

3.2 Interpretability Evaluation

We will evaluate interpretability based on the question categories from the XAI question bank, as they can be seen in Figure 1. The description of the abbreviations used in the table is provided in Appendix A. Using just performance metrics is not enough to provide answers to these questions and to understand explainability. Two algorithms might have the same F1-score; however, one may better understand the task than the other. Therefore, we will also partially make use of already available tools that are useful for this task.

The main tool that has been used for understanding models' predictions is LIME [14], a popular explainability tool that supports individual predictions for any black-box classifier for text or images. Several studies indicate that LIME can justify predictions with relevant evidence, including [15]. An explanation given by LIME is a local linear approximation of the model's behaviour. To do this, LIME perturbs the instance to be explained and learn a sparse linear model around it as an

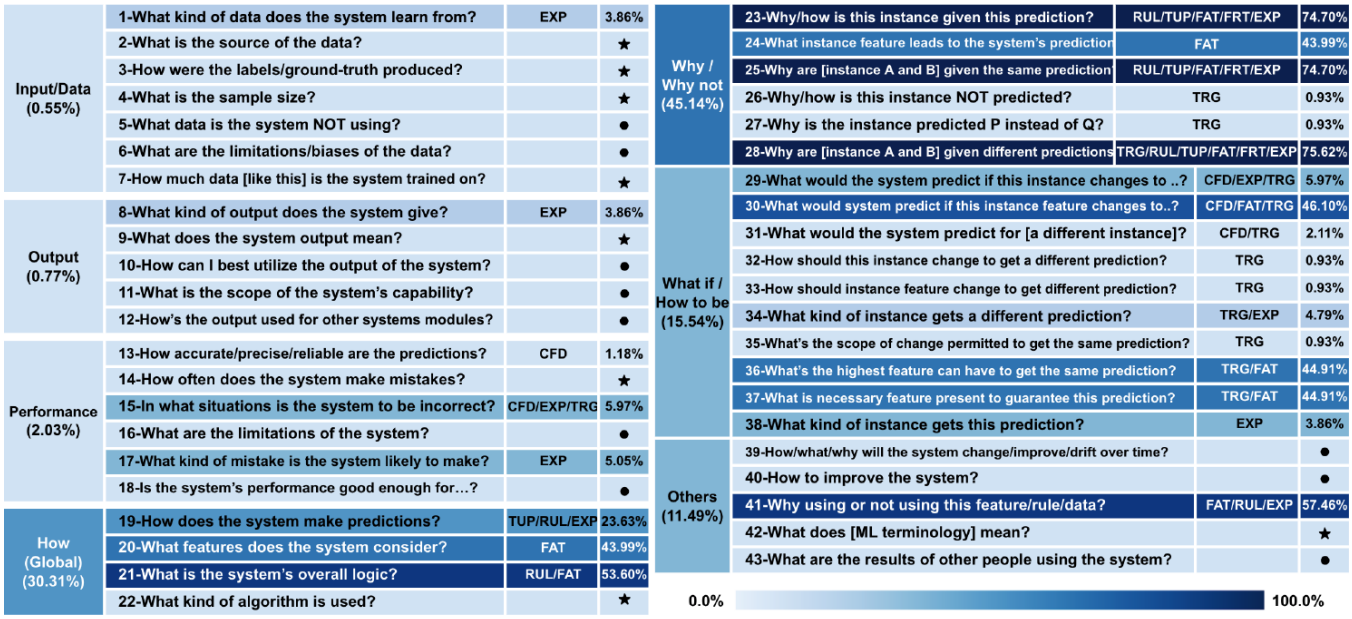


Figure 1: The questions in XAI Question Bank, heat-mapped by the estimated percentage (%) of NLP XAI studies attempting to answer them.(●: questions that cannot be answered by most NLP XAI studies; ⋆: questions that can likely be answered by the AI system's metainformation.), from [11]

explanation. This approximates the model well in the vicinity of a particular instance X, but not necessarily globally.

Another tool that was used to some extent is the Language Interpretability Tool (LIT) [16], a modular and extensible tool to interactively analyze and debug a variety of NLP models. The other experiments were run using code we created ourselves for each specific subtask.

4 Experimental setup

4.1 Dataset

MFTC Corpus

The data used to train and test our models comes from the Moral Foundations Twitter Corpus [6]. The dataset contains a collection of tweets from a broad range of topics chosen to maximise the variance in expressions of moral sentiment. The description of the corresponding subdatasets can be seen in Table 1.

Table 1: Moral Foundations Twitter Corpus Domains [6]

Corpus	Description
All Lives Matter	Tweets related to the All Lives Matter movement
Black Lives Matter	Tweets related to the Black Lives Matter Movement
Baltimore Protests	Tweets posted during the Baltimore protests against the death of Freddie Gray
2016 U.S. Presidential Election	Tweets posted during the 2016 U.S. Presidential Election
Hurricane Sandy	Tweets related to Hurricane Sandy, a hurricane that caused record damage in the United States
#MeToo	Tweets related to the #MeToo movement
Davidson Hate Speech	Tweets collected by Davidson et al. (2017) for hate speech and offensive language research

The tweets were labelled by several annotators (between three and eight) for their moral values according to the MFT [7]. The description of the five foundations that were used for this task is provided in Table 2.

Table 2: Definitions of moral foundations [17]

Foundation	Definition
Care Harm	This foundation is related to our long evolution as mammals with attachment systems and an ability to feel (and dislike) the pain of others. It underlies virtues of kindness, gentleness, and nurturance.
Fairness Cheating	This foundation is related to the evolutionary process of reciprocal altruism. It generates ideas of justice, rights, and autonomy
Loyalty Betrayal	This foundation is related to our long history as tribal creatures able to form shifting coalitions. It underlies virtues of patriotism and self-sacrifice for the group. It is active anytime people feel that it's "one for all, and all for one."
Authority Subversion	This foundation was shaped by our long primate history of hierarchical social interactions. It underlies virtues of leadership and followership, including deference to legitimate authority and respect for traditions.
Purity Degradation	This foundation was shaped by the psychology of disgust and contamination. It underlies religious notions of striving to live in an elevated, less carnal, more noble way. It underlies the widespread idea that the body is a temple which can be desecrated by immoral activities and contaminants (an idea not unique to religious traditions).

Data aquisition

The Moral Foundation Twitter Corpus (MFTC) dataset is publicly available online [18]. However, because it contains only the tweets' id's, the actual texts had to be fetched using the Twitter API. Data acquisition was initially challenged by the low availability of the original tweets, with only 49,9% of the total data publicly available at the time. Consequently, we requested the full dataset directly from the authors of the MFTC paper [6]. Experiments to follow were all conducted using the dataset in its entirety. With that, we hope to achieve an objective comparison to past work.

Annotations

Due to the subjective nature of these moral values, different annotators may label the same tweet differently. In order to assign a unique target vector to each of these tweets, we applied a majority vote, similar to the original paper [6]. This means that a tweet was labelled with a particular moral value, if and only if at least half the annotators agreed on that value. Where no such agreement was possible, tweets were labelled with a "non-moral" label.

Preprocessing

The first steps of preprocessing were to apply lower casing and punctuation and stopwords removal using spaCy [19]. Subsequently, we normalised commonly encountered social network syntaxes such as URLs, usernames and mentions and applied word segmentation and spelling correction using the specialised Ekphrasis package [20]. Emojis were also transformed to their corresponding words using the Python Emoji package [21]. For the data used to train the LSTM, lemmatisation was also performed in addition to all the previous procedures.

4.2 Model implementation and training

Three models have been evaluated during this research.

LSTM

The model architecture consists of several layers. First, there is the input layer followed by an embedding layer that was initialised with the weights from the pre-trained Glove6B embeddings with 100 dimensions. Next, there is the LSTM layer with 15 hidden nodes followed by a Global Max Pooling 1D layer. Finally, there is a dense output layer with a sigmoid activation function and size 11. The data that is fed to the model is processed using the Keras [22] tokeniser and padded with a maximum sequence length of 100.

BERT

We used *bert-base-uncased* [23] for both the tokenizer and the text classification model. This is a BERT model trained on lower-cased English text with 12 layers, a hidden-layer size of 768, 12 attention heads, consisting of 110M total parameters. For our task, a dense output layer of size 11 and no activation function has been added to the basic architecture. Before feeding the input to the model, the data is tokenised, padded with a maximum length of 64 and truncation is applied when necessary.

FastText

FastText is an open-source library, so the implementation was already provided. In order to use the library for training, prediction and evaluation, the data had to be transformed according to the fastText API requirements [24].

The training configurations for each of the three models has been provided in Appendix C. To obtain unbiased and consistent results, we shuffled the dataset and used 10-fold cross-validation.

4.3 Interpretability analysis

As mentioned in the methodology section, for evaluating the interpretability of the models, we will use the seven question categories from Figure 1. According to these categories, the following set of experiments has been performed[2].

Experiment 1: Performance

This experiment relates to the *Performance* category. There are three main questions that we try to answer here: (1) *"How accurate/precise/reliable are the predictions?"*, (2) *"What kind of mistake is the system likely to make?"* and (3) *"In what situations is the system to be incorrect?"*

For the first question, we have chosen to evaluate the performance using the precision, recall and macro-average F1-score. By treating the classes equally, the macro-average score is insensitive to any class imbalances in the data, compared to the micro-avg score, which aggregates the individual contributions of each class to compute the average (see Appendix B).

For the second one, we looked at the number of labels predicted, misclassification and conflicts errors per foundation.

For the third question, we looked at the percentage of test set errors by subdataset, when the models are trained on the whole MFTC data.

Experiment 2: Input data

This experiment relates to the *Input* category. The main question that we try to answer in this part is *"What kind of data does the system learn from"*. To do this, we ran experiments analysing the input of the model from different perspectives. We mostly looked at the class distributions, but we also considered the way texts were annotated (number of labels and conflicts within foundations).

Experiment 3: Embeddings visualisations

This relates to the *Other* category. The main question here is *"How does the model extract features from the data?"*[3].

The subquestion of this experiment will be answered by looking at the embeddings representations of the models. As we are working with raw sentences, in NLP we do not have predefined features. In order to extract features from data, word embeddings are used. These map each word in the vocabulary to a latent feature vector of multiple dimensions. The features are trained such that words with similar meaning are closer in the multidimensional vector space.

For Glove (used by LSTM) and FastText, we plotted the most frequent words by label using t-SNE [25] visualisation. The list of words has been constructed by counting how many times they appear in sentences corresponding to a certain label, and without considering stop words or dataset-specific nouns such as "sandy", "trump" or "baltimore" (see Appendix D). For BERT, we used the UMAP [26] visualisation on a sample of 10,000 input sentences.

[2] No experiments are directly assigned to the questions from the *Output* and *How* categories, as these can be answered either by the problem description or by the general conclusion of the results.

[3] This question is not included in the XAI question bank but is very similar to question 41.

Table 3: Model F1, Precision, and Recall Scores for Moral Sentiment Classification

Model	Metric	Average	Care	Harm	Fairness	Cheating	Loyalty	Betrayal	Authority	Subversion	Purity	Degradation	Non-moral
LSTM	F1	.50	.50	.55	.66	.59	.61	.37	.33	.17	.44	.47	.79
	Precision	.64	.74	.64	.67	.64	.82	.63	.50	.53	.47	.64	.70
	Recall	.44	.37	.48	.65	.55	.49	.26	.24	.10	.42	.37	.90
BERT	F1	.67	.73	.72	.76	.73	.72	.56	.63	.50	.54	.65	.84
	Precision	.70	.73	.72	.78	.74	.75	.58	.68	.57	.62	.69	.87
	Recall	.64	.73	.72	.75	.71	.68	.53	.59	.44	.48	.62	.81
FastText	F1	.57	.63	.57	.69	.59	.62	.40	.56	.44	.52	.49	.79
	Precision	.60	.65	.56	.70	.62	.66	.50	.57	.49	.57	.59	.70
	Recall	.55	.62	.57	.67	.57	.58	.33	.54	.39	.49	.43	.86

Experiment 4: Feature attribution
This relates to the *Why* category, the one that has received the most attention in scientific research.

We tackled this experiment by using the LIME tool. We tried to answer the following three questions: (1) *"What instance feature leads to the system's prediction?"*, (2) *"Why/how is this instance given this prediction?"*, and (3) *"Why are instance A and B given the same prediction?"*.

Given the impracticality of manually inspecting the whole dataset of 35,000 texts, a sampling procedure must be put in place. From the list of frequent words constructed in Experiment 3 (see Appendix D), we selected a representative word for each class and reviewed sentences containing these words. To ensure we explore the distribution as much as possible, we selected both entries annotated with the true corresponding class of the word and another unrelated class.

Experiment 5: Counterfactuals
This relates to the *What If* category. We will focus on answering the following two questions: (1) *"What would the system predict if this instance feature changes to ..?"*, and (2) *"What is a necessary feature to guarantee this prediction?"*.

The observations will also be made using the LIME tool. The first question will be tackled by replacing certain words with a word with an opposing meaning and observing in what way does the prediction of the model change. For the second question, we will be adding or eliminating words in order to see if they change the prediction or not. For complexity purposes, we mainly considered single labelled examples.

5 Results and Discussion

5.1 Performance

Looking at performance metrics is the first step in understanding a model's behaviour. These metrics quantitatively assess the prediction performance and inform us about the strengths and weaknesses of the model. Performance is to some extent correlated with the overall interpretability of the model, as it assures that the predictions are not random and there is some behaviour the model follows. Most explanations will be linked to the performance as this is the only way to observe the visible behaviour of the model in a quantifiable way.

Table 3 shows the aggregated precision, recall, and macro-average F1 scores for the three models, as well as the scores for each class. We can see that BERT is the best performing model with an average F1 score of 0.67. The second most performing is the FastText with a 10% lower score of 0.57 (10% lower than BERT), followed by LSTM, the worst performing model with a score of 0.5. Assuming that a better score indicates that a model that has learned more about the data, we expect BERT to be the one that can provide more stable explainability results. In contrast, for the other two, we expect to observe some random behaviour in their explanations.

Table 4: Predictions with both labels from the same foundation

Foundation	LSTM	BERT	FastText
care	4	12	15
fairness	6	3	13
loyalty	0	4	2
authority	2	6	9
purity	0	1	3
non-moral*	235	94	400

Note: non-moral represents predictions when any moral value is predicted together with the non-moral label. Test set contained approx. 3500 samples.

When we try to understand what kind of mistakes the system makes, we usually look at the confusion matrix. However, in the case of multi-label classification, constructing a confusion matrix is not a trivial task. One possibility is to count the true and predicted values every time they appear in their corresponding vectors, as performed on a similar task in [27]. However, this considers the labels individually and not as a group, so the true meaning and value are lost. Precisely for this task, an approach that could give some insight into the way models misclassify values is to examine the pairs of values from the same foundation. This is relevant because these pairs contain a virtue and a vice, so the model should be able to separate between two opposing values. Table 4 shows how many times moral values from the same foundation are predicted together. We can see that this behaviour is almost nonexistent for the five foundations. A similar result has also been observed for the misclassification of values within foundations (number of times the opposing value from a vice/virtue pair has been predicted). However, we observe a significant percentage of texts where non-moral is predicted together with one or more moral classes. This is conflicting behaviour as a particular text should either be moral on non-moral.

Table 5: Percentage (%) of texts assigned with a given moral value label, per dataset

Corpus	Care	Harm	Fairness	Cheating	Loyalty	Betrayal	Authority	Subversion	Purity	Degradation	Non-moral
MFTC	7.31	11.37	6.60	10.67	6.09	4.33	4.54	6.15	2.50	4.51	52.28
ALM	10.31	16.61	11.64	11.42	5.52	0.90	5.52	2.06	1.83	2.76	39.44
Baltimore	3.07	4.38	2.39	9.32	6.70	11.15	0.31	4.61	0.72	0.50	69.18
BLM	6.11	19.73	9.93	16.66	9.95	3.21	5.25	5.76	2.05	3.54	31.03
Davidson	0.18	2.83	0.08	1.27	0.84	0.84	0.41	0.14	0.10	1.37	96.49
Election	7.43	10.98	10.45	11.57	3.86	2.39	3.15	3.08	7.63	2.58	60.89
MeToo	4.26	8.95	8.08	14.16	6.66	7.57	8.58	18.07	3.58	19.45	36.03
Sandy	21.61	17.27	3.90	10.00	9.04	3.18	9.65	9.82	1.22	1.98	28.66

Table 6: Percentage (%) of test set errors by subdataset

Subdataset	LSTM	BERT	FastText
ALM	14.0	15.3	13.8
Baltimore	13.1	15.3	13.9
BLM	12.7	11.1	12.0
Davidson	3.3	2.9	3.1
Election	15.1	15.8	18.7
MeToo	21.5	22.6	22.2
Sandy	20.4	17.0	16.3

One more aspect to explore is to see in what situations is the system to be incorrect. More concretely, we look at what kind of data the errors come from. In Table 6 we can see the percentage of errors from the test set by subdataset on a model trained on the complete MFTC data. We observe that most errors come from the MeToo corpus. By looking at the class distribution of this dataset (in Table 5) we observe that subversion and degradation are the most common moral labels. However, considering the whole dataset, these are some of the most underrepresented classes, and on which the models also achieved the lowest F1-scores (see Table 3). Therefore, an explanation for this behaviour is the inability of the model to learn enough about these moral values to differentiate them from the others.

5.2 Input data

Models are built to understand the patterns in the data they are given, therefore their interpretability should be highly affected by the data they are trained on. It is worth mentioning that looking only at the data might not give any straight conclusions about explainability, but it will help better understand the results from other experiments.

A natural option for analysing the input data was to look at the class distributions. From Table 5 we can observe that the data is very unbalanced. For example, more than 50% of labels are non-moral, while some classes such as purity represent as low as 2.5% of the total data. Given this, it is expected that the model to perform worse on the underrepresented classes. Referring to the performance results, we can see that this is the case, subversion, purity, and betrayal being the classes with the lowest performance.

For a multi-label problem, it also makes sense to look at the number of labels they were assigned for each input text. We observe that most of the data (85.29%) have only one label assigned and that there are no texts assigned with more than five labels. It is expected that the models to perform worse on the predictions with multiple labels. This is confirmed by a follow-up experiment in which we observed a common behaviour among all three models. Around 84% of actual label vectors of size one are predicted with the correct size, this number dropping significantly to around 15% for vectors of size two. Vectors of size three had less than 5% accuracy, while sizes bigger than four are seldom predicted accurately.

Table 7: Percentage (%) of texts from MFTC dataset that are assigned both labels from a moral foundation

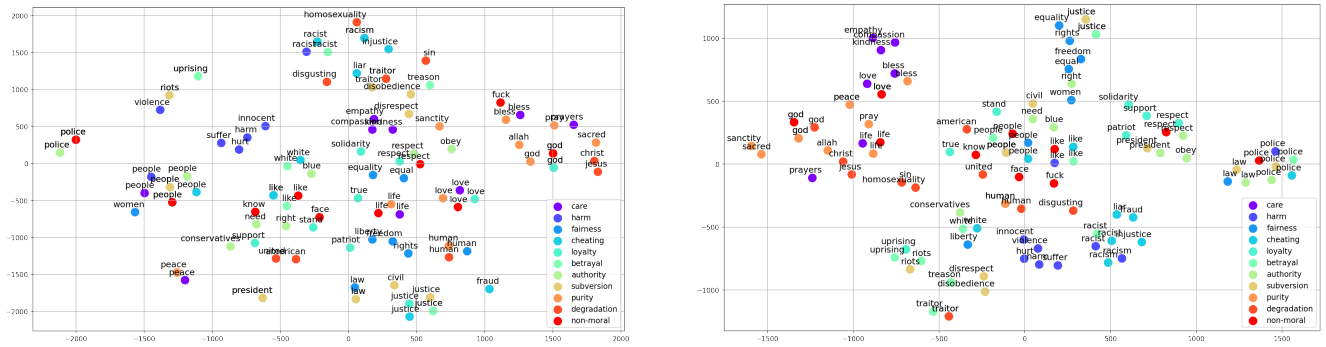
Care	Fairness	Loyalty	Authority	Purity	Non-Moral
0.41	0.27	0.09	0.13	0.06	3.47

Following the previous performance experiments, it would be wise to count the number of texts assigned with both labels from the same moral foundation. Such results can be seen in Table 7. While this is not a common issue for the five moral foundations, having almost 3.5% of the data annotated with both a moral label and the non-moral label is confusing. This happens when half of the annotators assign a label while the other half assigns another label, the majority vote keeping both labels as the correct ones. Interestingly enough, when removing this data, the same behaviour is still present. Therefore, no correlation can be made between this anomaly in the data and the predictions. Most probably, this happens due to the distribution of non-moral labels in the data. Having more non-moral values simply increases the probability of predicting the non-moral class.

5.3 Embeddings visualisation

Embeddings visualisation reveals the distribution of the features and similarities between certain words. While Glove and FastText learn traditional global word embeddings, BERT learns contextualised embeddings, meaning that a word has different representations depending on its context (sequence-level semantics).

In Figure 2 we have the two t-SNE plots of Glove and FastText embeddings for the most frequent words, grouped by moral value. Glove visualisation shows no clear delimitation of the words by their label. However, the presence of clusters is more notable for FastText. The clearest clusters are for classes like harm, care, fairness or loyalty, while



(a) Glove **(b) FastText**

Figure 2: 2D embeddings visualisations for Glove and FastText using t-SNE reduction

the most sparse classes are betrayal, authority and subversion (the classes with the lowest performance). For purity, the words also seem to be clustered together, but they are primarily words commonly used by multiple classes (such as "god", "life" or "human"). Probably, classes that have words with closer meaning are misclassified more often.

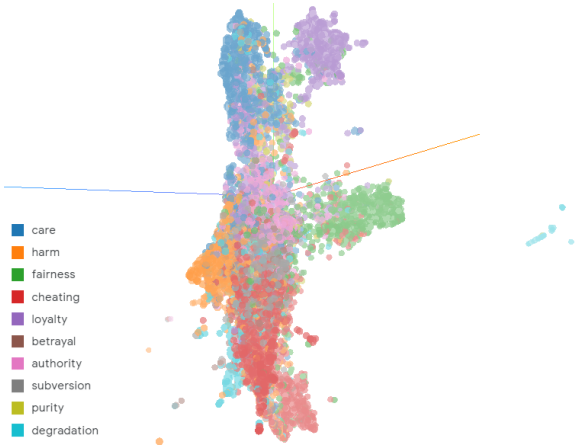


Figure 3: BERT 3D visualisation of the contextualized embeddings for moral classes using UMAP reduction

BERT's UMAP visualisation shows the presence of clusters for most of the labels except for subversion, purity and betrayal. This might be an explanation for why these classes achieve the lowest F1 scores. Moreover, the contextualised embeddings of BERT seem to be more meaningful than static embeddings (hence the difference in performance).

5.4 Feature attribution

With feature attribution, we try to assign systems' predictions to certain input features. An example of the output given by LIME can be visualised in Figure 4. Green words indicate loyalty, while blue words indicate care. The weights should indicate how much influence the words have on the prediction probability.

Regarding the models' comparison, BERT seems to be the most explainable of the three. It is clear that the contextualised embeddings provide more relevant information to the model as it is able to distinguish more subtle difference in meaning. Usually predicts correctly even if a top word is present and does not correlate any specific word with a specific class. BERT usually highlights more words (rather than one primary highlighted word for LSTM and FastText), meaning it does not base its prediction on a single word.

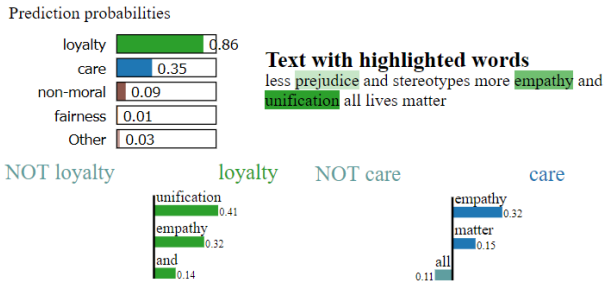


Figure 4: LIME example of BERT model correctly predicting both the *loyalty* and *care* labels

Although in few cases LSTM predictions were unexpected, looking at the word weights explained everything. Almost every time it highlights the frequent word, but sometimes it gives different predictions (in that case, the highlight is for NOT that class). Frequently, it predicts the class of the top word even if the actual label is other. It looks like usually one word is trained to be assigned to a class, opposed to BERT, where one word might be correctly assigned to multiple classes (e.g. "traitor" is always predicted as degradation even if the true label is betrayal).

FastText was the less explainable model with LIME, having many examples whose predictions could not be understood by just looking at the word weights. It usually had lower weights for the highlighted words than the other two models, and in cases where no precise class can be predicted, it highlighted words that provide no valuable meaning, such as stop words ("who", "at", "the") or commonly used words ("police", "god").

5.5 Counterfactuals

The counterfactual analysis tries to understand the behaviour of the model when part of the input is changed. To do this, the prediction on the initial input is compared with the output given from the altered input.

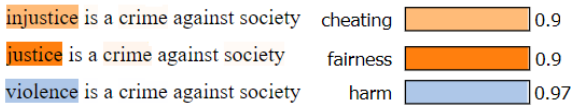


Figure 5: Example of models changing their predictions based on a single word

As we initially expected, changing ordinary words does not lead to a change in the prediction. Changes happen only when replacing expressive words that bring a lot of meaning to the sentence. However, negating the word with the most weight does not necessarily modify the outcome. Sometimes, it is the case that multiple words need to be replaced to alter the prediction. Even though the altered sentence might not have a coherent meaning, if the replacement is an opposite word with equal importance, it can radically switch the prediction (see Figure 5). Similarly, replacing with another common word also leads to that class prediction.

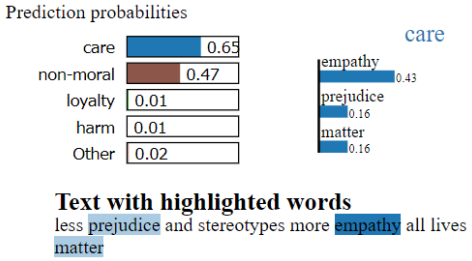


Figure 6: Example of BERT model changing the prediction when the word "unification" is removed

When having more than two labels, the behaviour is harder to predict and might be unexpected. By taking the same example from Figure 4 and removing the word "unification" we expect the model to still predict loyalty with a probability of $0.86 - 0.41 = 0.45$. However, we can see from Figure 6 that this is not the case, the model predicting *care* and *non-moral*. Moreover, the word prejudice, initially assigned to *loyalty* is now highlighted for the *care* class.

When it comes to comparing model predictions, the behaviour is quite similar. It often happens that the class changes to the non-moral class rather than another moral class. A difference that we noticed is that BERT is the only model that seems to understand negation properly. For the sentence "we stand against all forms of injustice" it is the only model to correctly predict *fairness*, while the other two predict *cheating* by considering just the individual word "injustice".

6 Responsible Research

6.1 Reproducibility

Reproducibility is an essential aspect of the research that needs significant attention. In order to provide verifiable results, the experiments that are run need to be reproducible by a third party entity. Moreover, ensuring easy reproducibility of the experiments also facilitates further research that will be conducted based on the obtained results. It is not uncommon that the progress of scientific research is sometimes hindered by the inability to completely understand previous work.

As we deal with a deep learning task, reproducibility is more challenging to achieve due to the non-deterministic characteristic of neural networks. Multiple factors affect the training process for a neural network, such as training data, hyperparameters or pretrained weights. Furthermore, given that all the aforementioned things are the same, there is still randomness given by the training methods (computation of the derivatives, weights initialisation, data shuffling). To overcome this issue, random seeds have been used to provide consistent results with minimal differences.

Other aspects that have been considered are modularity of the implementation and code quality (to provide an easy way to work with or change the code), the ease of model configuration (to be able to reproduce the experiment, you should be able to run it as easy as possible) and documentation (which is crucial in order to be able to understand and execute the provided code).

The details of the implementation and evaluation methods have been described in the experimental setup section of the report. The code repository will be made available on Github[4]. The experiments were run on the High-Performance Computing (HPC) Computing Cluster from TU Delft[5]. In order to be able to recreate the same Python environment locally, an *environment.yml* file containing all the required dependencies has been provided.

6.2 Ethical aspects

With the increasing impact of technology on our everyday life, critically assessing the ethical aspects of the research work that is being conducted becomes a necessity. Considering the irrevocable effects of a particular technology that has been introduced into society, it is the responsibility of the engineering community to think about the possible consequences of their work from the earlier design stages.

There are two main perspectives to be considered when performing an ethical reflection. A first perspective is to take into account the ethical aspects that arise while performing the actual research or development work. In our case, the most crucial factor that can be considered is the handling of personal data. As we are working with a dataset containing social media posts from Twitter, we need to make sure the users' privacy is respected. To do this, data should be anonymised such that no post can be traced back to the original user. Furthermore, the usage of consent forms can be employed. However, given the large number of users who need to be contacted, this option does not seem reliable.

[4]https://github.com/enricoliscio/nlp-for-values-CSE3000
[5]https://login.hpc.tudelft.nl/

The other perspective considers the ethical implications of the resulting technology. This implies viewing the broader scope of the technical project. In order to do this, both the primary and secondary stakeholders of the technology need to be identified. After this step, one needs to understand how these stakeholders are affected by the technology to further develop possible solutions that can maximise the benefits and minimise the harms.

Our research is mainly addressed to people working in the area of Artificial Intelligence and Machine Learning. However, given that the larger long-term motivation of the project is to enhance collaboration between humans and AL, the results of our work can also impact the general population. In essence, the outcome of our work should bring benefits for both the engineering community (the ones developing the technology) and the normal population (the ones that use the technology).

Firstly, we want to find how well NLP models perform on the task of moral classification. A favourable result in this direction would help us develop models that can observe social trends as they form. This way, we can understand what divides people, and we can react accordingly to overcome these divides.

Secondly, we are interested in the explainability of text classification models. Currently, the behaviour of most ML models is hard to understand. However, a better understanding of these models can lead to less bias and more accountable systems in which we can have more trust.

Finally, we can also identify some negative aspects that can arise in the work that we are conducting. The main problem is represented by the computational power that these models require for training and running. As we are progressively moving towards a more eco-friendly living environment, we should try to build and use systems that demand fewer resources.

7 Conclusions and Future Work

Moral values are abstract notions that ground our judgments and motivate our behaviour. They represent concepts we care about and are connected to fundamental human emotions and experiences. Estimating these values from online discourse represents an essential step towards creating value-aligned collaborative agents that better understand human decisions and actions.

In this paper, we performed a qualitative analysis of the interpretability of three models widely used in the NLP community: LSTM, BERT and FastText, on the task of estimating moral values from text. The main research question was to find whether the behaviour of these models can be explained when they are trained for moral value prediction. As there is no specific procedure for assessing interpretability, we had to establish a concrete methodology ourselves. We used the XAI question bank to guide our exploration of interpretability. According to this, we created five experiments that would allow us to rigorously answer the research question: (1) evaluating the performance of the models, (2) inspecting the input data, (3) interpreting embeddings visualisations, (4) exploring feature attribution using LIME and (5) investigating counterfactual predictions.

Our study revealed that the great unbalancedness of the moral classes in the MFTC data seems to affect all models' performance and behaviour. Although misclassification within moral foundations is negligible, we noticed that misclassification between moral and non-moral classes is more prominent. It has been observed that models also lack the ability to predict labels with more than two moral values accurately.

Our results also show that frequent and meaningful words that are also part of the Moral Foundation Dictionary generally have a very high impact on the outcome of the predictions. We think that the models might perform better on text distributions where certain distinguishable words and phrases appear frequently, rather than on corpuses with more diversified vocabulary. Therefore, it would also be wise to understand better how well the models generalise on other data.

When comparing the three models, BERT achieved the best performance with an F1-score of 0.67 and, according to our expectations, also seems to be the most interpretable model. Embeddings visualisations and LIME experiments showed that BERT is better at learning to estimate moral values, being able to differentiate words based on context and noticing subtle semantic particularities such as negation. As BERT provided the most promising results, it would make sense to further investigate this model's interpretability. It would be interesting to delve deeper into this model's architectural details and look over the attention mechanism. Performing more extensive hyperparameter tuning, testing different preprocessing and tokenisation strategies or using other explainability tools might further help in understanding BERT's behaviour.

8 Acknowledgments

The implementation of the NLP models used in this research has been made in collaboration with several other Bachelor students from Delft University of Technology, namely Alin Dondera, Andrei Geadau, Dragos Vecerdea and Florentin Arsene. We would also like to thank Associate Professor Morteza Dehghani from the University of Southern California for providing the full Moral Foundation Twitter Corpus dataset at request.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[3] Y. Belinkov, S. Gehrmann, and E. Pavlick, "Interpretability and analysis in neural NLP," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, Online: Association for Computational Linguistics, Jul. 2020,

pp. 1–5. DOI: 10.18653/v1/2020.acl-tutorials.1. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-tutorials.1.

[4] J. Graham, J. Haidt, and B. A. Nosek, "Liberals and conservatives rely on different sets of moral foundations.," *Journal of personality and social psychology*, vol. 96, no. 5, p. 1029, 2009.

[5] J. A. Frimer, *Moral foundations dictionary 2.0*, Dec. 2019. DOI: 10.17605/OSF.IO/EZN37. [Online]. Available: https://osf.io/ezn37.

[6] J. Hoover, G. Portillo-Wightman, L. Yeh, S. Havaldar, A. M. Davani, Y. Lin, B. Kennedy, M. Atari, Z. Kamel, M. Mendlen, *et al.*, "Moral foundations twitter corpus: A collection of 35k tweets annotated for moral sentiment," *Social Psychological and Personality Science*, vol. 11, no. 8, pp. 1057–1071, 2020.

[7] J. Graham, J. Haidt, S. Koleva, M. Motyl, R. Iyer, S. P. Wojcik, and P. H. Ditto, "Moral foundations theory: The pragmatic validity of moral pluralism," in *Advances in experimental social psychology*, vol. 47, Elsevier, 2013, pp. 55–130.

[8] O. Araque, L. Gatti, and K. Kalimeri, "Moralstrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction," *Knowledge-Based Systems*, vol. 191, p. 105 184, 2020, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2019.105184. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095070511930526X.

[9] T. Miller, *Explanation in artificial intelligence: Insights from the social sciences*, 2018. arXiv: 1706.07269 [cs.AI].

[10] Q. V. Liao, D. Gruen, and S. Miller, "Questioning the ai: Informing design practices for explainable ai user experiences," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–15.

[11] H. Shen and T.-H. '. Huang, *Explaining the road not taken*, 2021. arXiv: 2103.14973 [cs.CL].

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, *Bag of tricks for efficient text classification*, 2016. arXiv: 1607.01759 [cs.CL].

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, *"why should i trust you?": Explaining the predictions of any classifier*, 2016. arXiv: 1602.04938 [cs.LG].

[15] P. Lertvittayakumjorn and F. Toni, "Human-grounded evaluations of explanation methods for text classification," *arXiv preprint arXiv:1908.11355*, 2019.

[16] I. Tenney, J. Wexler, J. Bastings, T. Bolukbasi, A. Coenen, S. Gehrmann, E. Jiang, M. Pushkarna, C. Radebaugh, E. Reif, and A. Yuan, *The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models*, 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.15.

[17] MoralFoundations.org. (2021). "Moral foundation theory," [Online]. Available: https://moralfoundations.org/. (accessed: 27.06.2021).

[18] M. Dehghani, J. Hoover, G. Portillo-Wightman, L. Yeh, S. Havaldar, Y. Lin, A. M. Davani, B. Kennedy, M. Atari, Z. Kamel, and et al., *Moral foundations twitter corpus*, May 2019. [Online]. Available: https://osf.io/k5n7y.

[19] M. Honnibal. (). "Spacy," [Online]. Available: https://github.com/explosion/spaCy. (accessed: 27.06.2021).

[20] C. Baziotis. (). "Ekphrasis," [Online]. Available: https://github.com/cbaziotis/ekphrasis. (accessed: 27.06.2021).

[21] T. Kim and K. Wurster. (). "Emoji," [Online]. Available: https://github.com/carpedm20/emoji. (accessed: 27.06.2021).

[22] Keras. (). "Keras," [Online]. Available: https://github.com/keras-team/keras. (accessed: 27.06.2021).

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. (). "Bert base model (uncased)," [Online]. Available: https://huggingface.co/bert-base-uncased. (accessed: 27.06.2021).

[24] Facebook. (). "Fasttext," [Online]. Available: https://github.com/facebookresearch/fastText. (accessed: 27.06.2021).

[25] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[26] L. McInnes, J. Healy, and J. Melville, *Umap: Uniform manifold approximation and projection for dimension reduction*, 2020. arXiv: 1802.03426 [stat.ML].

[27] A. F. van Luenen, *Recognising moral foundations in online extremist discourse: A cross-domain classification study*, 2020.

A Explainable AI methods

FAT = Feature Attribution
Description: Highlight the sub-sequences in input texts
Typical question: How can we attribute the systems' predictions to input features?

TUP = Tuple/Graph
Description: explain model reasoning process with tuples/trees/ graphs
Typical question: How does the system use reasoning graphs to arrive at the answer?

CPT = Concept/Sense
Description: Convert to human interpretable concepts or terminologies
Typical question: What sense does the system's intermediate representation make?

RUL = Rule/Grammar
Description: Extract executable rules or logic for model decisions
Typical question: How can we explain the system's behavior with executable rules?

PRB = Probing
Description: Classify representation with specific diagnostic dataset
Typical question: What linguistic proper-ties does the system's representation have?

FRT = Free Text
Description: Use natural language to ex-plain model behavior
Typical question: How can we explain a system's decision using natural language justification?

EXP = Tuple/Graph
Description: Find most responsible training samples as explanations
Typical question: How can we trace the system's prediction back to the training sample(s)most responsible for it?

PSP = Projection Space
Description: Project dense vectors into low-dimensional space
Typical question: How can we project the system's high-dimensional representation to a human-understandable space?

B Performance metrics

The micro-average score is computed as follows:

$$precision_{micro} = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + FP_i}$$
$$recall_{micro} = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + FN_i}$$
$$F1\ score_{micro} = 2 * \frac{precision_{micro} * recall_{micro}}{precision_{micro} + recall_{micro}}$$

The macro-average score is computed as follows:

$$precision_{macro} = \frac{\sum_{i=1}^{C} precision_i}{C}$$
$$recall_{macro} = \frac{\sum_{i=1}^{C} recall_i}{C}$$
$$F1\ score_{macro} = 2 * \frac{precision_{macro} * recall_{macro}}{precision_{macro} + recall_{macro}}$$

C Training configurations

LSTM

- *epochs = 10*
- *batch size = 128*
- *learning rate = 0.01*
- *optimiser = Adam*
- *loss function = binary cross-entropy*
- *threshold = 0.4*

BERT

- *epochs = 4*
- *batch size = 32*
- *learning rate = 0.01*
- *optimiser = AdamW*
- *loss function = binary cross-entropy with logits*
- *dropout = 0.1*
- *threshold = 0*

FastText

- *epochs = 50*
- *learning rate = 0.03*
- *threshold = 0.25*
- *embedding dimensions size = 100*

D Most frequent words by label

care: 'love', 'compassion', 'god', 'kindness', 'bless', 'peace', 'people', 'empathy', 'life', 'prayers'

harm: 'people', 'hurt', 'violence', 'police', 'racist', 'racism', 'harm', 'suffer', 'like', 'innocent'

fairness: 'justice', 'equality', 'rights', 'human', 'freedom', 'equal', 'people', 'liberty', 'women', 'law'

cheating: 'injustice', 'fraud', 'racist', 'racism', 'people', 'justice', 'police', 'liar', 'white', 'like'

loyalty: 'solidarity', 'love', 'support', 'patriot', 'justice', 'god', 'true', 'stand', 'respect', 'uprising'

betrayal: 'traitor', 'riots', 'people', 'police', 'racist', 'uprising', 'like', 'justice', 'treason', 'white'

authority: 'respect', 'obey', 'law', 'police', 'blue', 'right', 'need', 'people', 'president', 'conservatives'

subversion: 'disobedience', 'people', 'civil', 'disrespect', 'police', 'law', 'traitor', 'riots', 'justice', 'president'

purity: 'god', 'life', 'sacred', 'sanctity', 'love', 'human', 'bless', 'peace', 'pray', 'allah'

degradation: 'traitor', 'disgusting', 'sin', 'god', 'american', 'united', 'christ', 'jesus', 'human', 'homosexuality'

non-moral: 'like', 'love', 'people', 'face', 'god', 'police', 'respect', 'fuck', 'know', 'life'