Performance analysis of the state-of-the-art NLP models for predicting moral values

Andrei Geadău[1] , **Pradeep Murukannaiah**[1] , **Enrico Liscio**[1]

[1]Faculty of Computer Science, TU Delft

Abstract

Moral values are instrumental in understanding people's beliefs and behaviors. Estimating such values from text would facilitate the interaction between humans and computers. To date, no comparison between NLP models for predicting moral values from text exists. This paper addresses this by comparing LSTM and more novel models such as BERT and fastText to evaluate their capabilities for predicting moral values. Twitter Corpus, a collection of 35000 Tweets containing relevant recent political and social events, is chosen for this purpose. The results show that novel solutions outperform long-established ones. BERT is proven to be the best model for this task, but long training times hinder its practicality. By contrast, fastText offers similar performance while being orders of magnitude faster.

Keywords: Values, Ethics, NLP

1 Introduction

Our culture, traditions, laws, and experiences led individuals to create systems of values based on the standards of good and bad. Moral values allow us to understand the distinction between desirable and undesirable actions, thoughts, opinions, and behavior. Moral Foundation Theory (MFT) [1] narrows down these abstract philosophical concepts into a subset that can be evaluated for scientific and practical reasons. It proposes five foundations, each consisting of two opposite labels: care/harm, fairness/cheating, loyalty/betrayal, authority/subversion, and purity/degradation [2]. Additionally, the eleventh label, non-moral, is often added as a fallback, signifying the open-ended nature of this modular classification.

The importance of injecting moral values into computers is self-explanatory: it allows machines to determine and motivate one's behavior in relation to society. Possible future advancements can be seen in search engine recommendations, social network filters, chatbots, personal assistants, and text-based decision-making research [3].

Related work

Many recent works attempt to capture and model morality from text. A prime example is Rezapour et al.'s work [4], which captures morality and stance from Twitter posts to measure social effects. It uses the Baltimore dataset [5], a collection of Tweets from the violent 2015 protests. The same dataset is used in Mooijman's study [5] on moralization in social networks, which correlates the posts with the intensity of the protestors' violence.

Introduced by Hoover et al.[2], the Moral Foundation Twitter Corpus (MFTC) aims to combat the shortage of annotated datasets for moral value classification. It consists of 35.000 Tweets, spread equally among seven different heterogeneous domains that were relevant at the time of writing: All Lives Matter (ALM), Black Lives Matter (BLM), the Baltimore protests, the 2016 Presidential election, hate speech and offensive language, Hurricane Sandy, and #MeToo [2]. Labels for the tweets represent the set of five universally agreed moral values, which constitute the Moral Foundations Theory.

Problem statement

The previously-mentioned works have not treated novel models for classifying moral values. A complete comparison study would include transformer language models [6] and text classification libraries, which are shown to outperform the ones present in Hoover et al.'s paper [7], although this remains heavily dependent on the dataset, as Ezen-Can points out [8]. One limitation of Hoover's work is the poor performance of the chosen models, which makes readers think that moral value classification remains unfeasible. Long Short Term Memory (LSTM), the best-performing model presented, only achieves an F1 score of .41 (sd .02).

Contribution

Here we perform an extensive comparison of state-of-the-art NLP models in estimating moral values from text. In particular, Google's BERT [9] and Facebook's fastText [10] are used. A modified version of the LSTM model is included in the final version for consistency purposes. It is used as a benchmark for evaluating the classification between the three models. This paper aims to critically examine the performance and training time of moral value classifiers to determine which model is best suited for this task.

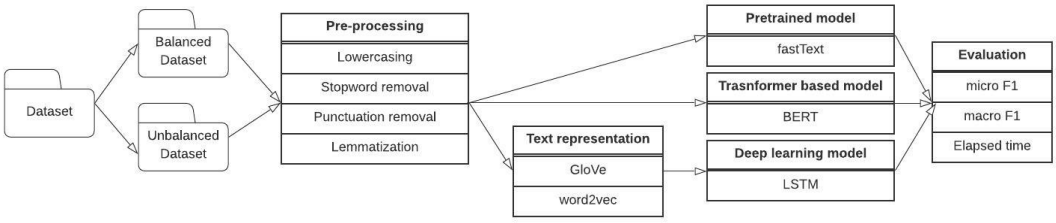

Figure 1: Overview of methodological approach.

Structure

This paper is organized as follows: Section 2 presents in detail the methodological approach: choice of the dataset, fairness issues, data pre-processing, choice of word embeddings, details about the three models used, and metrics used. Section 3 highlights the experimental setup. Section 4 covers the results of the experiments. Section 5 addresses the ethical implications of performing such a comparison on user's data. Last but not least, Section 6 informs the reader about future improvements and recommendations.

2 Methodology

The previous section introduces the reader to moral values and illustrates the importance of moral value classification from textual inputs. In this chapter, the research methodology explains how the study is performed.

The research methodology in this work is mainly drawn from Maslej et al.[7], who provide a clear framework for evaluating multi-label, multi-class NLP models. They establish the importance of gathering the data, pre-processing it, splitting it into testing and training data, selecting the models, optimizing their performance by tuning hyperparameters. Figure 1 provides an overview of the methodological approach. The remainder of this section provides an in-depth rationale for each point.

2.1 Models

The three models used in this paper are chosen upon careful consideration. While providing details about the models themselves is outside the scope of the paper, a brief overview is required to understand our reasoning.

LSTM

Long short-term memory (LSTM) [11] represents a type of recurrent neural network (RNN). A more complex structure aims to overcome the network's internal memory loss for long input data, also known as the vanishing gradient problem [12]. The simple addition of a Forget Gate, determining which information is relevant and adjusting its flow accordingly, makes it suitable for classifying sentences or paragraphs. Figure 2, taken from [7], illustrates the improvement of LSTM over a standard recurrent neural network by preserving context information through the entirety of the network.

The inclusion of LSTM in a state-of-the-art comparison may come as a surprise. It is certainly not considered a state-of-the-art model, being published in 1997 and researched in various scientific publications on the topic. Instead, it is included for consistency purposes. Due to its wide use, it can be used as a baseline for comparison with other models to evaluate their relative increase in performance, rather than the absolute value, as the latter is low due to the complexity of the classification problem. Hoover et al.'s paper lists LSTM as the best-performing model, achieving better results than Support-vector machines.

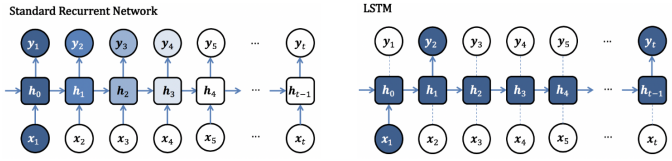

Figure 2: Illustration of the vanishing gradient problem in a standard recurrent network versus long-short term network, from [7].

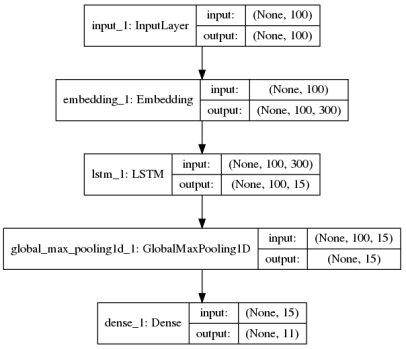

Figure 3: Architecture of LSTM model for multi-label classification.

fastText

FastText, an open-source library developed by Facebook AI Research, is argued to be "on part to deep learning classifiers in terms of accuracy, and orders of magnitude faster for training and evaluation" [10]. One can use it to learn word representations and text classification, the latter being relevant for this study. It combines state-of-the-art concepts employed by the NLP community. The "bag of words" (BoW) model of representation and n-gram decomposition of words allows for the classification of words that infrequently appear in the training data.

	Yahoo		Amazon Full		Amazon polarity	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
char-CNN	71.2	1 day	59.5	5 days	94.5	5 days
VDCNN	73.4	2h	63	7h	95.7	7h
fastText	72.3	5s	60.2	9s	94.3	10s

Table 1: Comparison between fastText and deep-learning models, from [13].

FastText is included in this paper due to its excellent trade-off between accuracy and training times. Joulin [10] claims that it achieves results just shy of the best deep learning models, but orders of magnitude faster training times. Table 1, taken from an official blogpost [13], highlights the results.

BERT

BERT (Bidirectional Encoder Representation for Transformers) [9] represents Google AI Language's state-of-the-art model at the time of writing. Its novelty consists of the attention mechanism, enabling it to process deep bi-directional relations between words within a sentence rather than performing a simple left-to-right iteration. A single output layer is then required in order to fine-tune the model. Previous works acknowledge BERT's potential as it turns out to be the best-performing model within a series of comparisons with other deep learning models. Malsej's et al.'s experiments [7] only shows a minor increase in overall F1 score, but this can be attributed to the lack of complexity in data. It is interesting to determine how this difference evolves given an objectively more difficult dataset.

Numerous variations of BERT exist and are worth taking into consideration. On the one hand, XLNet and RoBERTa both represent retraining of BERT and are shown to be capable of significantly outperforming it in certain circumstances [14]. On the other hand, DistilBERT [12] aims to speed up the computation at the cost of accuracy. The decision was to pursue BERT as it provides the most general, relevant, and straightforward solution. It also allows reinforcing findings of other relevant literature, such as [7].

2.2 Dataset

The comparisons between the models mentioned in the previous section were performed using MFTC, from [2].

The reasoning behind using MFTC is further detailed. Primary inclusion criteria represent the possibility to classify the data into abstract moral values. Popular sentiment analysis datasets have been taken into consideration ([14], [15]), but fail to achieve this because they can be reduced to binary classifications due to the non-ambiguous labels, which defeats the point of this work. Recent works, however, are promising. The Morality Machine [16] uses the same Moral Foundation Theory [2] labels, but the consensus was that 18,959 entries are insufficient for a fair comparison between the chosen models. The benefit of using MFTC is that it contains more entries than all related works while also being more complex: it has seven categories that share little context between each other. Therefore, it better reflects the ever-growing demands of value classification.

Gathering the data proved to be a challenging methodological obstacle. Only 49.9% of the original Tweets could be fetched using the API provided by the authors [17]. The rest had been either deleted by the creators or banned by Twitter, which comes as no surprise considering the sensitive messages they carried. Figure 2 shows the distribution per category. Missing half of the initial dataset, including the entirety of two categories (Davidson and MeToo), would make the comparison unfair. The remaining option was to contact the authors of the paper, who were kind enough to provide the entire dataset.

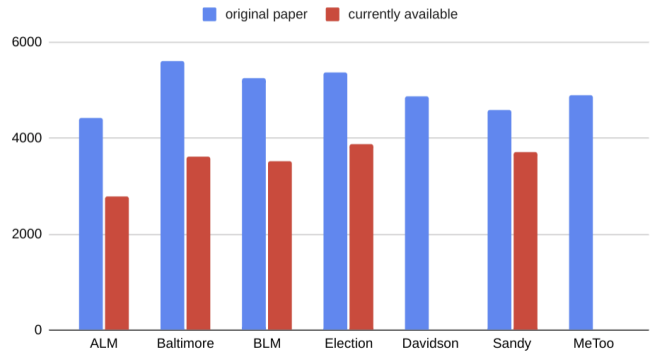

Figure 4: Tweets still available for retrieval using the public API.

2.3 Pre-processing

A quick look at examples immediately reveals particularities of text that harm the classification models: internet jargon, emojis, overuse of punctuation symbols, a mixture of lowercase and upper-case words, platform-specific symbols such as '#' or '@', phone numbers, or emails. Polamuri [18] established common methods that can be applied: converting text to lowercase, removal of personal identifiers (phone numbers, usernames, email addresses), removal of # symbol, removal of emojis, removal of stopwords, punctuation, or lemmatization. With an overwhelming number of possible combinations, the choices had to be reduced to a reasonable number for comparison. Four strategies, varying in complexity, are considered in this work. Table 2 highlights techniques employed by each of them. The averaged F1 score of seven LSTM classification is used to determine the pre-processing method used throughout this work.

Strategy	0	1	2	3
Only lowercase	✘	✔	✔	✔
No personal identity	✘	✔	✔	✔
Remove # symbol	✘	✔	✔	✔
No Emojis	✘	✘	✔	✔
No Stopwords	✘	✘	✘	✔
No punctuation	✘	✘	✔	✔
Lemmatization	✘	✔	✔	✔

Table 2: The four pre-processing strategies, varying in complexity.

2.4 Word Embeddings

LSTM is different from the other two models because it requires pre-trained word vectors to learn textual associations between words carrying similar meanings. Word Embeddings are vector representation of text, where each word maps to a set of real valued vectors in a pre-defined N-dimensional space. FastText uses its own pre-trained set word vectors, so there is no need to cover it in this section explicitly. Similar reasoning applies to BERT. Because of its attention mechanism, it is capable of understanding word associations within the text.

Glove [19] and Google word2vec [20] are suitable options that apply different unsupervised learning techniques on a variety of large datasets. On the one hand, Word2vec makes use of two different architectures, combining the CBOW's ability to take into consideration infrequent phrases and Skip-gram's fast training time. The final result has a large dimensionality of 300. On the other hand, Glove combines window-based methods, which also help to take into consideration rare phrases, but optimizes training times by keeping track of 'how frequently words co-occur with one another in a given corpus' [19]. The outputs are vectors with different dimensionalities of 50, 100, 200, and 300. Both methods have been used in relevant deep-learning, especially [21].

2.5 Metrics

This work uses standard metrics to evaluate the quality of the multi-label classifiers. Micro and macro F1 scores are reported for each of the experiments. In addition, the training time is measured for each experiment, as, in the case of indistinguishable gains, this will most likely be the determining factor in real-word applications. For completeness, this section illustrates the mathematical formulas for each of the scores, computed using the open-source library Scikit-learn [22].

	p	**n**
p′	True Positive	False Negative
n′	False Positive	True Negative

Table 3: Confusion matrix.

For mutli-label classification, each class c_i is individually taken into consideration. TP_i, FP_i, FP_i and FN_i are defined similarly to binary classification, with the exception that all c_j where $i \neq j$ are treated as a negative class (see Table 3). Therefore, the metrics can be calculated as such:

$$Precision_{micro} = \frac{\sum_{i=1}^{\|C\|} TP_i}{\sum_{i=1}^{\|C\|} TP_i + FP_i} \quad (1)$$

$$Recall_{micro} = \frac{\sum_{i=1}^{\|C\|} TP_i}{\sum_{i=1}^{\|C\|} TP_i + FN_i} \quad (2)$$

$$F1\ score_{micro} = 2 * \frac{Precision_{micro} * Recall_{micro}}{Precision_{micro} + Recall_{micro}} \quad (3)$$

$$Precision_{macro} = \frac{\sum_{i=1}^{\|C\|} Precision_i}{\|C\|} \quad (4)$$

$$Recall_{macro} = \frac{\sum_{i=1}^{\|C\|} Recall_i}{\|C\|} \quad (5)$$

$$F1\ score_{macro} = 2 * \frac{Precision_{macro} * Recall_{macro}}{Precision_{macro} + Recall_{macro}} \quad (6)$$

While both micro and macro F1 scores provide valuable insight into the classification performance, the latter is more beneficial in our case, as it is insensitive to the imbalance of the classes and treats them all as equal. Nevertheless, the two should be correlated most of the time.

Finally, the training time is measured by training the models on a personal computer with the following configuration: 4.00Ghz Intel Xenon (8 CPUs), 32 GB RAM, Nvidia Quadro P2200 5GB.

3 Experimental Setup

The setup for the experiments is presented in Tables 4 - 6. The hyperparameters were chosen to prevent overfitting and reduce the bias, which was presented in Subsection 4. Admittedly, the hyperparameters have not been exhaustively tested, as the number of combinations grows exponentially and the research has been performed within a limited time window. Further optimization can certainly be applied. Nevertheless, the chosen set-up is considered to reflect the models' capabilities of predicting moral values accurately.

Additionally, fairness was further improved by using k-fold cross-validation with random shuffling on both the pre-processed dataset (see Subsection 2.3) and the 'Balanced dataset' (from Subsection 4). The vast majority of data studies [23] show that the number k = 10 represents a suitable parameter, and it is also used in both works on which this one is based upon.

Hyper-Parameters	Values
Epochs	[3, 5, **10**]
Activation	[**sigmoid** , relu]
Batch size	[32, 64, **128**]
Optimizer	[**Adam**]

Table 4: Hyper-parameters used for LSTM. Chosen values are bold.

Hyper-Parameters	Values
Epochs	[10, **50**, 100]
Learning rate	[0.01, **0.03**, 0.05]

Table 5: Hyper-parameters used for fastText. Chosen values are bold.

Hyper-Parameters	Values
Epochs	[2, 3, **5**]
Batch size	[**16**, 32, 64]
Optimizer	[**AdamW**]
Dropout	[0.05, **0.1**, 0.02]

Table 6: Hyper-parameters used for BERT. Chosen values are bold.

4 Results

Dataset bias

In observational studies, there is a potential for bias caused by discrepancy in data. Figure 5 hints at the presence of bias in MFTC. As can be observed, 48% of the entire dataset is considered non-moral after applying majority vote. This leads to better training for the non-moral subset, which, in turn, harms the classification process. Table 7 further supports the discrepancies in training for the different labels. By selecting a random fold out of the possible ten during the training phase, the discrepancy in the F1 score of the classifiers on the non-moral and other labels becomes striking.

	ALM	Baltimore	BLM	Election	Davidson	Sandy	#MeToo
Subversion	91	257	303	165	7	451	874
Authority	244	17	276	169	20	443	415
Cheating	505	519	876	620	62	459	685
Fairness	515	133	522	560	4	179	391
Harm	735	244	1037	588	138	793	433
Care	456	171	321	398	9	992	206
Betrayal	40	621	169	128	41	146	366
Loyalty	244	373	523	207	41	415	322
Purity	81	40	108	409	5	56	173
Degradation	122	28	186	138	67	91	941
Nonmoral	1744	3848	1583	2502	4509	1313	1618
Total	4424	5593	5257	5358	4873	4591	4891

Figure 5: Frequency of Tweets per Foundation Calculated Based on Annotators' Majority Vote. Image provided at request.

Class	BERT	fastText	LSTM
fairness	0.76	0.74	0.62
non-moral	0.73	0.81	0.76
purity	0.0	0.44	0.0
degradation	0.0	0.34	0.0
loyalty	0.94	0.48	0.24
care	0.71	0.53	0.41
cheating	0.72	0.53	0.39
betrayal	0.0	0.35	0.0
subversion	0.0	0.34	0.0
authority	0.0	0.45	0.0
harm	0.54	0.46	0.39

Table 7: F1 score of a randomly selected fold for each of the three models. Illustrates the massive differences per category.

A possible solution to this problem is to balance the dataset by downsampling the Tweets labeled as non-moral. This has been achieved by reducing the number of non-moral Tweets to match the closest category. We will hereafter refer to this as 'Balanced dataset', and it will be used in the final comparison between models, alongside the dataset obtained from Subsection 2.3, which we will refer to as "unbalanced dataset".

Pre-processing Strategy

No significant differences were found between the four strategies in terms of F1 score when applied to LSTM and fastText. Tables 8 and 9 reveals this by illustrating the micro and macro F1 scores trained using the four different strategies. Despite the small differences, a slight increase in F1 score can be observed as complexity increases. Several factors play a role in determining the effects of this trend. One reason is the removal of the excessive amount of meaningless data that is fed into the model, which is generally seen as a factor strongly related to poor performance. Optimizing the models, in particular increasing the number of epochs to the point of overfitting, may also explain the small differences. The intrinsic benefits of using a certain strategy are balanced by overtraining. Table 10 is a good illustration of how overtraining decreases the differences in dataset. Using five epochs has a best-worst difference of 0.3, whereas ten epochs display a marginal 0.1.

Strategy	0	1	2	3
micro F1	0.62	0.63	0.63	0.62
macro F1	0.44	0.42	0.42	0.43

Table 8: Averaged 'micro' and 'macro' F1 score of LSTM afer applying the four different strategies. Model trained in 10 ephocs using Glove[19] word embedding.

Strategy	0	1	2	3
micro F1	0.64	0.66	0.65	0.66
macro F1	0.52	0.51	0.52	0.51

Table 9: Averaged 'micro' and 'macro' F1 score of fastText afer applying the four different strategies.

Strategy	0	1	2	3
micro F1	0.57	0.59	0.6	0.58
macro F1	0.27	0.3	0.32	0.28

Table 10: Averaged 'micro' and 'macro' F1 score of LSTM afer applying the four different strategies. Model trained in 5 ephocs using Glove[19] word embedding.

The F1 score of BERT is expected to mimic the slight increase of the other two models. The experiments have not been performed due to time constraints. Overall, Strategy 3 gives the best results and is used for all experiments presented in this paper. The fact that the experiments are unable to demonstrate a correlation between pre-processing complexity and F1 score further provides confidence in using this strategy.

Word Embeddings

Figure 6 displays the averaged 'micro' and 'macro' F1 scores using the five different word embeddings considered. This figure is quite revealing in several ways. First, the directly proportional relation between dimensionality and accuracy seems to hold for the vectors of norm 200, 100, and 50. However, Glove-300 fails to deliver improved results compared to Google's word2vec, scoring an average micro F1 score of 0.62 whereas Google 0.64.

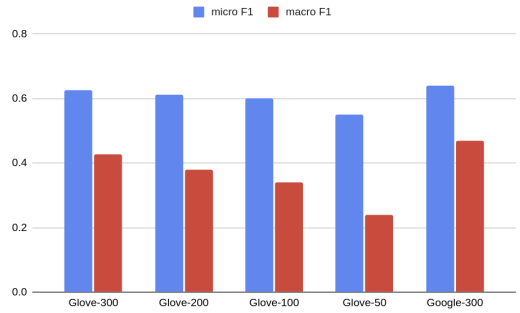

Figure 6: Averaged 'micro' and 'macro' F1 score of LSTM with the five different word embeddings. All experiments were trained in 10 ephocs.

However, one major limitation of Google word2vec represents the training time. It decelerates the training process by a factor of 4 because of the 10GB of space required, almost ten times larger compared to its Glove-300 counterpart. Figure 7 shows the accuracy per time unit of the two word embeddings, justifying the discrepancy in accuracy per time unit. Since training time does not represent a strict requirement in our case, Google word2vec is selected as the word embedding for LSTM in the rest of the experiments. The reader should be aware of this tradeoff, as, in most circumstances where training time is more critical than obtaining the absolute best results, it is possible to select Glove-300 as the word embedding of choice.

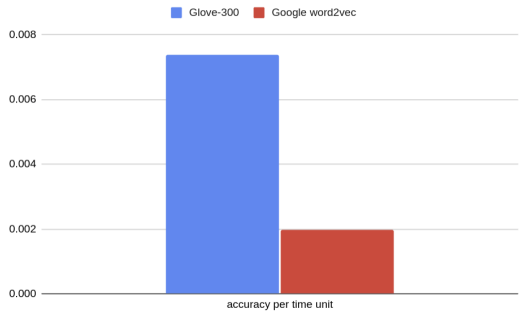

Figure 7: Glove-300 and Google's word2vec accuracy per time unit.

Balanced dataset

The results obtained from evaluating the three models on the unbalanced dataset are shown in Figure Figure 9. This figure is quite revealing in several ways. First, we can see that LSTM resulted in the highest value of micro F1 score, 0.64. However, the macro F1 score of 0.47 hints towards the undesired result of artificially increasing this score by overfitting. In contrast to LSTM, BERT has a more negligible difference between the two scores. While its micro F1 score is 0.05 smaller compared to LSTM's, the macro F1 score is 0.09 higher. Fasttext falls short compared to both models, achieving 0.46 micro and 0.39 macro F1 score.

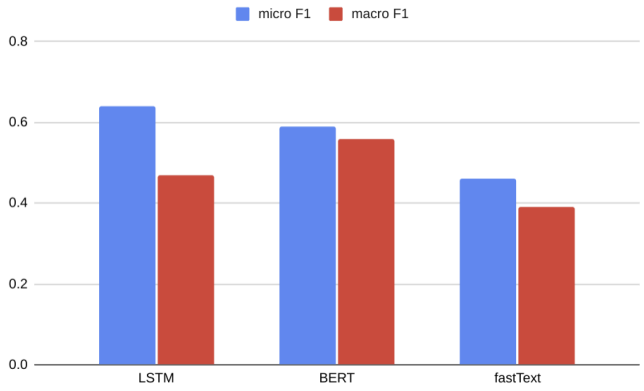

Figure 8: Averaged 'micro' and 'macro' F1 score of LSTM, BERT and FastText on balanced data.

Unbalanced dataset

The results of evaluating the unbalanced dataset are summarized in Figure 9. LSTM achieves a micro F1 score of 0.64, with a macro score of 0.47. BERT manages a 0.7 micro F1 score, and 0.6 macro F1 score. FastText accomplishes a 0.65/ 0.51 score.

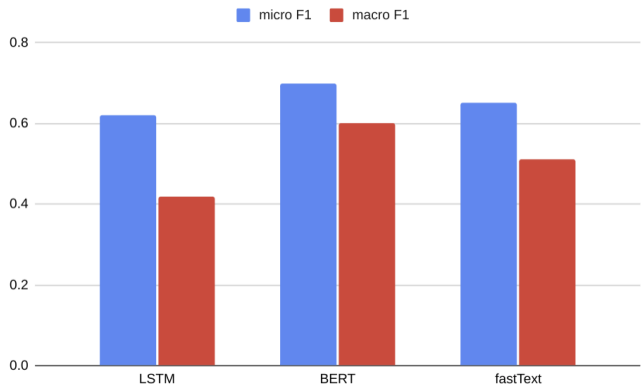

Figure 9: Averaged 'micro' and 'macro' F1 score of LSTM, BERT and FastText on unbalanced data.

The most surprising aspect of these results is the decrease of BERT and FastText when evaluated on the balanced dataset. While LSTM remains virtually unchanged on the two datasets, BERT has a 16% increase, while FastText achieves a substantial 30% improvement.

Training time

As can be observed from Figure 10, the difference in training times is considerable between the three models. FastText remains faithful to its name, delivering results orders of magnitude faster compared to BERT or LSTM. Whereas fastText completes the classification process in 85 seconds, the other models take considerably longer. LSTM, running on CPU takes 2583 seconds, and BERT requires 8740 seconds when using a powerfull GPU.

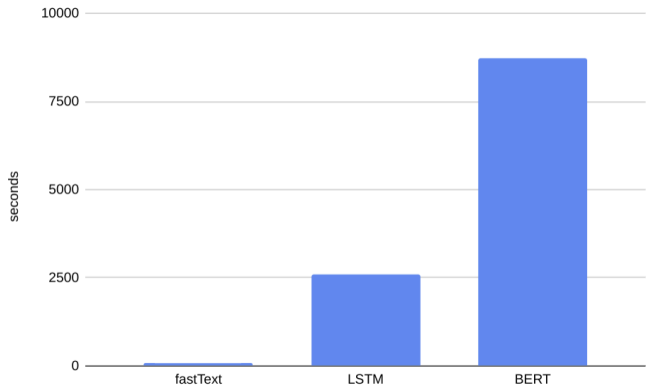

Figure 10: Training time (seconds) for the three models.

Moral foundations

Figures 11 and 12 provide the breakdown of the three models when classifying the five moral foundations, rather than all eleven moral values. The single most striking observation to emerge from this is the clear dominance of BERT, which achieves an 0.17 increase over LSTM on the balanced dataset, and 0.07 on the unbalanced one. This is a rather surprising outcome. It shows that BERT 's attention mechanism is capable of determining contextual meaning, but falls short of determining whether the moral value is a virtue or vice. No significant increase in LSTM and fastText was found compared with the previous experiments.

5 Responsible Research

The research described in this paper is conducted following responsible values and principles in order to ensure high academic standards. To the best of our knowledge, all results have been transparently and objectively reported, without any human intervention that could fabricate, falsify or misrepresent data. In exceptional cases where the results do not coincide with the expectations, multiple tests are performed to verify the genuineness of the experiments. An explanation for the difference between expectations and reality is included whenever necessary.

The legality of this work is upheld by the Twitter Developer authorization received. Twitter users have been granted access to all intellectual property present in this paper. Despite this, strict measures are used to ensure that no individual can be mentally or physically harmed as a result of any direct or

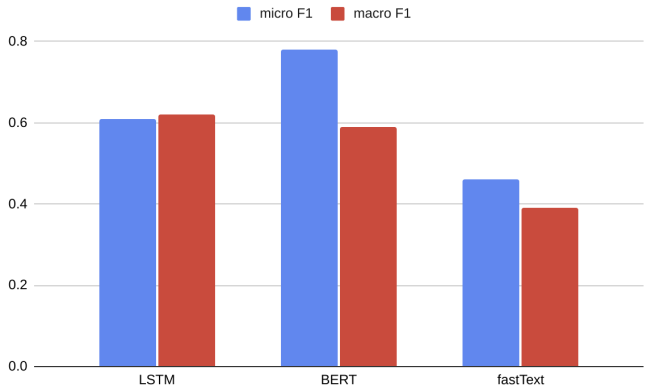

Figure 11: Averaged 'micro' and 'macro' F1 score of LSTM, BERT and FastText on balanced data, classification performed on moral fundations.

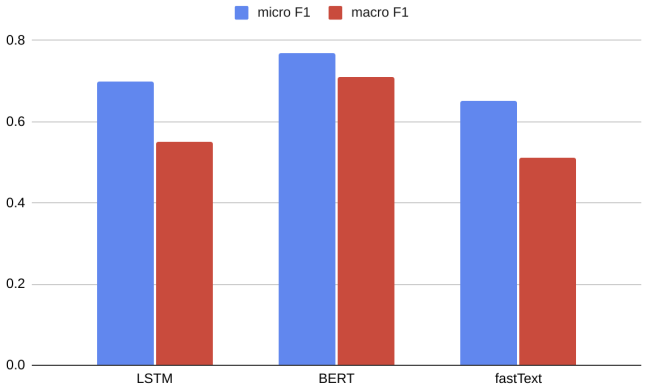

Figure 12: Averaged 'micro' and 'macro' F1 score of LSTM, BERT and FastText on unbalanced data, classification performed on moral fundations.

indirect action resulting from performing this research. Confidentiality and anonymity have been achieved by removing any identifiers (username, email) that can reveal information about a specific person. No Tweet's content is included in the paper.

The decision not to publish the dataset, which has been obtained at request from the authors of the Twitter Corpus dataset, has an adverse impact on the reproducibility of the work. While the source code is available on GitHub [1], under an MIT license, the interested reader is requested to contact the authors to acquire the dataset. Detailed information is provided to run the experiments presented in this paper. The reader should also be aware that all models are non-deterministic, which implies that no two simultaneous experiments will result in the same output. However, slight variations are presented within the paper.

Automating the understanding of moral values from text by creating better models for moral value classification repre-

[1] https://github.com/enricoliscio/nlp-for-values-CSE3000

sents a delicate ethical consideration that has both advantages and disadvantages. On the one hand, improvements in this area can benefit numerous people by enabling them to better interact with their devices and filter unwanted content from the internet. On the other hand, authoritarian governments or injurious entities can take advantage of the novel discoveries to censor, ban or hide content that may harm an individual's freedom of speech. While this paper is written hoping that the discoveries aid individuals, the dangers should not be neglected.

6 Discussion

The results will doubtless be scrutinized, but some immediately reliable conclusions can be drawn. Firstly, Ezen-Can's claim [8] proved to hold: a general-purpose model for moral value classification cannot be established, as all three models presented in this paper are heavily dependent on the training set. It was hypothesized that BERT would outperform LSTM, which, in turn, would result in comparable results to fastText. This is indeed the case for the majority of experiments, but not on the all of them.

One unanticipated finding was the effect of balancing the dataset. Balancing the dataset proved a major factor, if not the only one, causing the different levels of performance. LSTM outperforms all other models when evaluated on the balanced dataset, while BERT is the clear winner on the unbalanced one. It is difficult to explain why LSTM benefits significantly more from this bias compared to the other two models. Regardless, the comparison between the two datasets highlights the importance of adequately pre-processing the data, and encourages the reader to think twice about the circumstances in which the models are run.

These findings have important implications for moral value classification. Overall, the results help us to understand that, in majority of cases, BERT and fastText are believed to be the better solutions for classifying moral values. This statement is in accordance with the results by [7], and [10], both using balanced datasets. However, it may be the case that the reverse implication does not hold: if the dataset is unbalanced, then LSTM is the chosen solution. In that case, further investigations should be made in order to determine the most suitable model.

7 Conclusion

Moral values represent the systems of values based on the standards of good and bad. Moral value classification from text provides a better understanding of human sentiments, emotions and thoughts. This work presents a comprehensive comparison between state-of-the-art models for moral value classification. LSTM, BERT and fastText are evaluated on Twitter Corpus, a collection of 35000 Tweets specifically annotated for this task. A clear winner cannot be identified because of the high variations in performance on different training sets. On balanced datasets, BERT achieves the best performance, but the same is generally no true for unbalanced ones.

Then, the question remains when to use a model over another, considering that the dataset is balanced, such as we have in the first experiments. If training time is the main requirement, fastText is the most suitable option. It achieves decent results given the extremely short training time on the CPU. It is in part with the other two models, even outperforming LSTM in the first comparison. BERT and LSTM, on the other hand, are extremely slow for any real-time application. BERT is trained in roughly 3 hours using an expensive, professional-grade setup that costs in excess of 1000 $ at the time of writing. LSTM is no better either in this regard. While the training in this paper has been performed on CPU rather than GPU, it is expected that the 40 mins reported in the previous section can be reduced to about 10-15 minutes, a significant amount nevertheless. The decision to use one over the other then comes down to the expectations of the study. If spending two additional hours for a 0.06 increase in F1 score is an objective, then BERT should undoubtedly be taken into consideration.

8 Future Work

While this research has achieved its objective, it has also thrown up many questions in need of further investigation. For instance, it is not within the scope of this paper to examine the significant variations in F1 score between the two datasets, despite the fact that it offers a better understanding of the model's capabilities. Constantinescu's and Dondera's papers [24] [25] cover explainability and transferability of the same models considered here, using the same dataset. The reader is suggested to consult these papers for a more in-depth understanding of these experiments. Another question that remains unanswered has to do with four pre-processing strategies considered in this paper (see Table 8). Once again, no reasoning is provided for the insignificant difference between the strategies, as the results does not seem to coincide with the expectations.

Possibly the most significant limitation of this study represents the small number of deep learning models considered. For LSTM, possible extensions that can be adressed in future studies include the addition of backwards propagation phase, or using a hybrid architecture with a convolutionary neural network in order to better reflect word context. For BERT, the alternatives mentioned in Subsection 2.1 are certainly worth taking into consideration. One cannot deny that the inclusion of these models would make the comparison more exciting but, due to practical constraints, this paper cannot provide such ample comparison.

9 Acknowledgements

I would like to express my gratitude to Prof. Pradeep Murukannaiah and Enrico Liscio, my research supervisors, for their constructive suggestions during the planning and development of this research work. My grateful thanks are also extended to my colleagues, Florentin Arsene, Ionut Constantinescu, Alin Dondera, Dragos Vecerdea and Cheyenne Slager.

References

[1] Ain Simpson. *Moral Foundations Theory*, pages 1–11. Springer International Publishing, Cham, 2017.

[2] Joe Hoover, Gwenyth Portillo-Wightman, Leigh Yeh, Shreya Havaldar, Aida Mostafazadeh Davani, Ying Lin, Brendan Kennedy, Mohammad Atari, Zahra Kamel, Madelyn Mendlen, et al. Moral foundations twitter corpus: A collection of 35k tweets annotated for moral sentiment. *Social Psychological and Personality Science*, 11(8):1057–1071, 2020.

[3] Ola Leifler and Henrik Eriksson. Automated text-based analysis for decision-making research. *Cognition, Technology Work*, 14:1–14, 06 2011.

[4] Rezvaneh Rezapour, Saumil H. Shah, and Jana Diesner. Enhancing the measurement of social effects by capturing morality. In *Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 35–45, Minneapolis, USA, June 2019. Association for Computational Linguistics.

[5] Marlon Mooijman, Joseph Hoover, Ying Lin, Heng Ji, and Morteza Dehghani. When protests turn violent: The roles of moralization and moral convergence, 11 2017.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[7] Viera Maslej-Krešň'akov'a, Martin Sarnovsk'y, Peter Butka, and Krist'ına Machov'a. Comparison of deep learning models and various text pre-processing techniques for the toxic comments classification. *Applied Sciences*, 10(23):8631, 2020.

[8] Aysu Ezen-Can. A comparison of lstm and bert for small corpus, 2020.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[10] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics, April 2017.

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

[12] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6:107–116, 04 1998.

[13] Armand Joulin Tomas Mikolov Piotr Bojanowski, Edouard Grave. fasttext, 2016.

[14] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[15] Datafiniti. Consumer reviews of amazon products, 2019.

[16] Livia Teernstra, Peter van der Putten, Liesbeth Noordegraaf-Eelens, and Fons Verbeek. The morality machine: Tracking moral values in tweets. In Henrik Boström, Arno Knobbe, Carlos Soares, and Panagiotis Papapetrou, editors, *Advances in Intelligent Data Analysis XV*, pages 26–37, Cham, 2016. Springer International Publishing.

[17] Morteza Dehghani. Moral foundations twitter corpus, 2019.

[18] Sharmila Polamuri. 20+ popular nlp text preprocessing techniques implementation in python, 2020.

[19] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[20] Google. word2vec, 2013.

[21] Viera Maslej-Krešňáková, Martin Sarnovský, Peter Butka, and Kristína Machová. Comparison of deep learning models and various text pre-processing techniques for the toxic comments classification. *Applied Sciences*, 10(23), 2020.

[22] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[23] Charles Menguy. Choice of k in k-fold cross-validation, 2012.

[24] Ionut Constantinescu. Evaluating interpretability of state-of-the-art nlp models for predicting moral values, 06 2021.

[25] Alin Dondera. Estimating transferability of state-of-the-art models in predicting moral values, 06 2021.