

Moral Style Transfer

Stanford CS224N Custom Project
Staff Mentor: Angelica Sun

Matthew Radovan
Department of Computer Science
Stanford University
mradovan@stanford.edu

Sasankh Munukutla
Department of Computer Science
Stanford University
sasankh@stanford.edu

Mustafa Khan
Department of Computer Science
Stanford University
mkhan7@stanford.edu

Abstract

Moral reframing is the process of framing a statement in a way that is consistent with an individual’s moral values to garner support from that individual. Prior studies have shown morally-reframed messages help in persuasion on a range of topics [1], from environmental protection to reducing political polarization. We developed a moral classifier and a generator to morally reframe texts, which is the first NLP approach to do so. Specifically, we developed a moral classifier that can predict the underlying morals in a text using a BART-based architecture and a moral reframer that can perform moral reframing of an input text based on a target set of morals using a variational autoencoder architecture. Our best moral reframer model achieves F1 scores of 0.998 on our test set and a BERTSCORE (using BART) of 0.972, thus being able to perform moral reframing while preserving content.

1 Introduction

It has been well studied that political polarization has been increasing globally [1] and antipathy between liberals and conservatives is at its worst in the US [2]. Highly negative views have more than doubled since 1994: 43% of Republicans and 38% of Democrats now view the opposite party in strongly negative terms [2]. Moral foundations theory [3] has shown that human morals determine our political viewpoints, proposing innate and universally available psychological systems present in human beings regardless of culture or upbringing. There are five categories of these moral foundations, each of which can be framed as a virtue (or vice): (1) care (harm), (2) fairness (cheating), (3) loyalty (betrayal), (4) authority (subversion), and (5) purity (degradation). Moral reframing [1], whereby a position an individual would not normally support is framed in a way that is consistent with that individual’s moral values, has been shown to be an effective means for political communication and persuasion and can reduce this polarization.

Prior studies [1] have used humans to morally reframe messages to help in persuasion on a range of topics including views of environmental protection and same-sex marriage. As far as we know, our work is the first attempt to use NLP techniques to morally reframe texts. We focus on COVID-related topics for our texts as there is no prior human effort in morally reframing COVID messaging at large scale and COVID is currently one of the most pressing problems in society, so our work will maximize impact and can help fight the ongoing pandemic.

2 Related Work

Prior NLP work has focused on just identifying the underlying morals in a text. In particular, Lin et. al. used textual feature extraction with an LSTM based architecture to identify the moral foundations in tweets [4], showing that using background knowledge, in particular through entity linking to entities in Wikipedia, improved performance and achieved F1 scores comparable to a human annotator. Horne et. al. created a database of news articles to study misinformation, NELA [5], and used NLP techniques to extract relevant features, including moral foundations features using the methods developed by Lin et. al. Araque et. al. [6] also built on the work by Lin et. al, creating a moral lexicon of lemmas and used logistic regression models trained on features extracted from the lexicon to achieve better performance compared to Lin et. al. While Araque et. al. showed that extracting moral features combined with a simple model can achieve state of the art results, we aim to utilize an approach based on a transformer encoder to create a more nuanced and sensitive classifier. No prior work has utilized such an approach to find the underlying morals in a text, which our project develops for our moral classifier in Section 3.

As far as we know, this is the first attempt to use NLP techniques to morally reframe texts. Chawla and Yang [7] created a model that achieves formality style transfer, preserving the content of an informal sentence while making it semantically and grammatically correct. They developed an unsupervised approach and achieved strong results without requiring the use of a paired dataset of (informal, formal) texts, which is helpful for our task as we do not have a paired dataset. Specifically, they used cyclic GANs by training a single encoder-decoder pair in both directions, which teaches the decoder to translate the encoder’s learned embeddings to either of the two styles. In addition to

training the encoder-decoder, the paper uses a language model-based discriminator that is trained on unpaired data. It uses two language models (on the formal data and informal data separately), and its output is a score of how formal its input text is. In the supervised setting, Chawla and Yang introduce a third loss (MMI), which maximizes the mutual information between the translated output and the ground truth to ensure the translated text’s content is preserved, which we develop a proxy for given our task is unsupervised (see Section 3).

3 Approach

We develop a **moral classifier**, which determines the underlying morals in a text and a **moral reframer**, which is a generative model that takes a text input and a target set of morals and generates a text output that matches the target morals while preserving the content of the original text.

3.1 Moral Classifier

For our classifier model, similar to Chawla and Yang [7], we used an architecture based on the encoder of the BART model pretrained on the CNN-DM dataset [8], imported using the HuggingFace Library [9]. As our task is multi-label classification (a single text may map to multiple morals), we modified the BART model to add a custom sequence classification head. Specifically, inspired by BERT [10], we added a linear layer with output size equivalent to the BART encoder output’s hidden size followed by a tanh activation, followed by another linear layer with output size equal to the number of possible classes. We took the resulting state of the first token in that linear layer’s output sequence as the input for our linear layers. This resulted in 10 final outputs: 1 for each possible moral type (5 moral types · 2 vice/virtue directions). Finally, as this is a multi-label classification problem, we used binary cross entropy to evaluate the moral category outputs individually. Note that as opposed to the original BERT paper, we did not use a custom [CLS] token to start the input token, as our model is exclusively for classification and so the start sequence token (which was still included) is implicitly the [CLS] token.

Additionally, for compatibility with the generator (discussed later), we trained the classifier on one-hot encoded token sequences rather than discrete token sequences. To do so, we replaced BART embedding layer with a linear layer, without bias, whose weights were the stacked BART embeddings of each token in order. In doing so, the linear layer applied to a one-hot encoded token vector is equivalent to the corresponding embedding layer applied to the sequence of discrete tokens. Further mentions of this approach will describe it as a *pseudoembedding layer*, as it allows for either one-hot encoded tokens or softmax probabilities as input. When softmax probabilities are its input, it simply creates a weighted embedding based on the input probability distribution. This approach is inspired by Xu, Ge, and Wei [11] to approximate the token decoding process.

3.1.1 Baseline

Our baseline is a simple long short term memory (LSTM) model architecture with a hidden size of 128, embedding dimension of size 32, a dropout layer to prevent overfitting, and three linear layers prior to the linear output of size 11. Similarly to our classifier model, we used binary cross entropy as the loss after sigmoiding the model outputs. The hyperparameters chosen were manually optimized, with larger embedding dimensions and hidden layer sizes not showing significant changes in performance. We chose this baseline because prior to 2017, LSTMs were the standard architecture for NLP tasks. Additionally, it is a simplified version of the architecture Lin et al. used for a similar task [4], and requires no pretraining.

3.2 Moral Reframer

Our generative model was designed as a variational autoencoder, generally following the architecture proposed by Chawla and Yang [7]. So, both the encoder and decoder portions of the autoencoder were from the BART model pretrained on CNN and Daily Mail data [8]. We adopted a variational autoencoder over a GAN, as it is much less computationally expensive, and our classifier is not a true discriminator.

3.2.1 Pretraining

In order for our generative model to learn to preserve content, we explored:

1. **Pretraining:** This involved first training the generative model with the same target morals as that of the input text for 10 epochs so the model learns to effectively reconstruct the input text as an output. Then, after pretraining, the model is then trained with a random set of target morals (which are different from the input text’s morals) to teach it to learn how to perform moral transfer.
2. **No Pretraining:** Explored for comparison, by directly training the generative model with a random set of target morals for 20 epochs.

3.2.2 Generator Input Strategies

As we are using multiple target morals as input, rather than having a binary translation direction, we break from Chawla and Yang [7]. Specifically, rather than using a specific input token at the beginning of the generated text to determine the direction of translation, we defined three strategies to incorporate the target morals as input to the model:

1. **Feed Encoder:** The target morals were included as special tokens at the end of the encoder input for the model to learn the proper latent space representation of the input text given the target morals. Unlike Chawla and Yang [7], moral tokens were appended rather than prepended to avoid shifting the source text.
2. **Feed Decoder:** Similarly to the "feed encoder" approach, the target morals were included as special tokens at the end of the decoder input for the model decoder to learn how to properly combine the latent space representation of the original text with the target morals.
3. **Contextual Injection:** The target moral vector was concatenated to each vector in the hidden output sequence returned by BART encoder model. Because the output of the encoder has shape (sequence length, hidden size), we concatenate the moral feature vector to each context vector to achieve a new sequence encoding of size (seq len, hidden size + moral vector size). Because these new encodings are simply concatenated vectors that each have different meanings, they are then fed through a single linear layer to combine them into a new, coherent encoding vector. By injecting the moral feature vectors into the encoded state of the input sequence, the decoder learns to utilize the modified latent space representation to produce text with the target morals.

3.2.3 Frozen Model

When training our generator, we explored **freezing the encoder’s parameters**, **freezing the decoder’s parameters**, and **freezing neither** to allow the whole model to train. We initially had to freeze at least one of the encoder or decoder due to memory and speed constraints, as freezing neither the encoder nor decoder resulted in too many trainable parameters. After obtaining access to more powerful GPUs (Tesla V100s), we also experimented with fully trainable models.

3.2.4 Content Loss Metrics

To achieve a proxy for the MMI approach Chawla and Yang use in the supervised setting, we developed our own content loss metrics inspired by average embedding distance as discussed by Clark, Celikyilmaz, and Smith [12]:

1. **Embedding Space L_2 -normalised Euclidean distance:** We take the tokenwise mean of the input text sequence’s (I) and output text sequence’s (O) embedding. We normalize I, O by dividing them by their magnitudes so they are unit vectors I', O' . We then find the L_2 norm of the difference of two: $\|I' - O'\|_2$, which indicates the level of similarity between the content of the input and output sequences. This is directly inspired by Clark, Celikyilmaz, and Smith [12].
2. **Latent Space L_2 -normalised Euclidean distance:** We apply the same procedure as in the embedding space L_2 -normalised Euclidean distance. Rather than defining I, O to be the input and output sequences’ average embedding vector, we define them to be the average latent space vector by averaging the encoder’s last hidden state across the sequence length for both input and output sequences. This approach was inspired by the idea that the latent space may represent the content without being significantly affected by the moral framing.

3.2.5 Training

The generator was pretrained as discussed in Section 3.2.1. To randomly generate moral targets to train our generator, we generated random moral vectors such that those target vectors were not the exact same as the morals of the input text and competing morals (e.g. care and harm) could not both be set as targets.

To fully train the generator model, two losses were defined: the *moral loss* and the *content loss*. To calculate the moral loss, the sequence of softmax probabilities predicted by the generator model were fed to the inputs of the classifier model. The pseudoembedding layer discussed in Section 3.1 now applies because conversion of softmax probabilities to discrete tokens is not differentiable. So, feeding softmax probabilities into the linear pseudoembedding layer allowed backpropagation to occur uninterrupted. Then, the moral loss was defined as the binary crossentropy loss between the input target morals and the classifier’s predicted morals.

The content loss was calculated as defined by the variations in Section 3.2.4. Again, in order to compute the embeddings and encoder’s last hidden state from the generator’s output softmax probability sequence, a pseudoembedding layer was created based on the encoder’s embedding layer, and the output sequence’s input embeddings to the encoder were calculated using this pseudoembedding layer. Finally, the moral loss and content loss were summed to calculate the final loss.

4 Experiments

4.1 Data

We used the NELA 2020 dataset [13], constrained to COVID-related articles. The dataset provides 699,803 COVID-related articles across 493 different sources, where each article includes its title and content. We manually filtered out news sources that we did not immediately recognize as intentional sources of information (e.g. removing The Onion, while keeping Breitbart). Only article titles were kept due to the long length of article contents being computationally infeasible to work with. In addition to the dataset, the NELA project [5] provides a library to automatically extract a set of features from text, including moral features as defined by Moral Foundation Theory [3]. The moral features of each article’s title were extracted using this library - the "primary dataset." In addition, for further classifier experiments, the moral features of each article’s contents were extracted - the "content moral dataset," to further asses our model’s ability to identify the morals of a wider text based of just the headline.

Because a significant number of the moral feature vectors were 0, only articles with nonzero moral feature vectors were kept. This resulted in a final dataset of approximately 53,000 article titles and their moral features. Elements of the moral feature vector were floats ranging from 0 to 1, so to frame our data as a multi-label classification problem, we set all nonzero values in the moral feature vector to 1, indicating presence of the moral. We used a train-validation-test split of 80%-10%-10% to obtain our train, validation and test set:

Dataset	# articles
Train	42540
Validation	5318
Test	5317

Table 1: Data split

To preprocess our dataset for our baseline, we stripped all non-alphanumeric characters, removed all words that occurred only once, and tokenized the resulting words. For our classifier model, we tokenized the texts based on HuggingFace’s BART tokenizer pretrained on the CNN-DM dataset [9].

4.2 Evaluation method

4.2.1 Classifier

We trained our baseline and finetuned our classifier model on our training sets and tuned hyperparameters like the learning rate on the validation set. We then evaluated the best models for both on the corresponding test sets. We considered the metrics of accuracy (as defined by % of output vector that matched their target vectors perfectly) and F1 scores, both macro and micro, using scikit-learn [14]. Macro F1 score is based on the average classwise F1 score, while the micro F1 score is calculated using overall true/false positive/negative statistics, without regard to class. By using the macro F1

score, we were able to compare with Lin et. al [4], who achieved a macro F1 score of 0.7396 across five moral foundations on tweets (with no accuracy or micro F1 reported). However, we did not train on Lin’s dataset because our dataset and task of news titles is semantically different from tweets, so they are related but not parallel problems.

4.2.2 Generator

For our experiments, we evaluated our models on the same three metrics as above: accuracy, macro and micro F1 scores on the validation set to gauge performance. We then evaluated our best model from these experiments for both the moral transfer ability and the content consistency of our generator. We used BERTSCORE [15] to evaluate content consistency, where scores range from 0 to 1. It uses transformer language models (in our case, using BART rather than BERT) to compute a similarity score between each token in the candidate sentence and each token in the reference sentence; the similarity is computed using contextual embeddings rather than exact matches. A value closer to 1 signifies higher content consistency. To evaluate the moral transfer ability, we computed the F1 scores of the target input morals as compared to both the classifier predicted morals of the output text and the NELA feature morals of the output text.

4.3 Experimental details

All training and evaluation was done on NVIDIA Tesla V100s and NVIDIA GTX 1080Tis, depending on machine availability.

4.3.1 Classifier

We trained and evaluated our classifier and baseline on both our primary dataset and our content moral dataset. The classifier was trained for 20 epochs and the baseline for 20 epochs. The classifier had a learning rate of 1e-5 and the baseline had a learning rate of 1e-3. Learning rates were tuned using PyTorch Lightning’s learning rate finder [16]). Both used batch sizes of 32. Each epoch of the classifier took 20 - 30 minutes depending on the GPU being used.

4.3.2 Generator

We trained and evaluated our generator on our primary dataset. The pretrained models were pretrained for 10 epochs and all models were directly trained for 20 epochs using batch sizes of 8 and a learning rate of 1e-6, which we tuned using PyTorch Lightning’s learning rate finder[16]. Each epoch took between 45 - 60 minutes depending on the GPU being used.

4.4 Results

4.4.1 Classifier

Model	Accuracy*	Micro	Macro	Subversion/Authority	Cheating/Fairness	Harm/Care	Betrayal/Loyalty	Degradation/Purity
Baseline (LSTM) - Title Morals	0.978	0.989	0.947	0.993	0.779	0.994	0.975	0.975
Classifier (BART-based) - Title Morals	0.995	0.998	0.994	0.998	0.986	0.999	0.996	0.994
Baseline (LSTM) - Content Morals	0.068	0.642	0.239	0.525	0.169	0.719	0.490	0.232
Classifier (BART-based) - Content Morals	0.109	0.677	0.477	0.643	0.156	0.709	0.554	0.321

Table 2: Moral Classifier Validation Results (all values are F1 scores unless noted *)

Model	Accuracy*	Micro	Macro	Subversion/Authority	Cheating/Fairness	Harm/Care	Betrayal/Loyalty	Degradation/Purity
Baseline (LSTM) - Title Morals	0.974	0.983	0.927	0.989	0.753	0.994	0.975	0.975
Classifier (BART-based) - Title Morals	0.997	0.998	0.994	1.0	0.982	0.999	0.999	0.992
Baseline (LSTM) - Content Morals	0.057	0.631	0.232	0.507	0.154	0.713	0.487	0.201
Classifier (BART-based) - Content Morals	0.094	0.674	0.463	0.607	0.327	0.712	0.540	0.285
Lin model (different dataset)	N/A	N/A	0.739	0.721	0.748	0.798	0.803	0.6285

Table 3: Moral Classifier Test Results (all values are F1 scores unless noted *)

Note that the Lin model is run on a different dataset and is simply included for comparison. First, comparing our validation and test results, we see that our classifier achieves similar performance on both, indicating our model has generalized well. As our classifier has a more complex (BART based) architecture compared to our baseline LSTM, it makes sense that our classifier has improved performance over the baseline for both title and content morals. Additionally, it makes sense that both our baseline and classifier performed significantly better on the title moral prediction dataset compared to the content moral dataset, as the moral framing of the content may be only subtly linked

to the title text. Notably, our classifier’s predictions of content morals from the title has some F1 scores comparable to those of Lin’s model on Tweet data. This is likely because the pretrained BART model includes background knowledge, so it’s able to leverage that background knowledge to develop correlations between topics in article titles and the morals they’re most likely associated with, even with lack of explicit association. It is interesting to note that our classifier especially outperforms the baseline LSTM on macro F1 Score and several moral categories F1 scores for content morals, which highlights its ability to generalize better on harder tasks.

We ultimately used the classifier trained on the title morals for the moral loss when training our generator given it had the strongest performance and is more relevant to our task of generating headlines as output, compared to the content morals dataset.

4.4.2 Generator

Classifier Accuracy	Classifier Micro	Classifier Macro	NELA Acc	NELA Micro	NELA Macro	BERTSCORE
0.989	0.998	0.998	0.006	0.064	0.059	0.972

Table 4: Moral reframer evaluation results with best model on test set
(all values are F1 scores unless noted)

We ran a total of at least 28 experiments across our categories (2 options for pretraining, 3 options for feeding, 3 options for freezing, 2 content loss functions) as seen in Table 8 in the appendix, validating on the validation set after every epoch. If we observed loss was constant for the first 10 epochs or all validation metrics stayed at 0, we killed experiments as indicated with a * to preserve computational resources due to the high epoch time. In general, pretraining seemed to help, which makes sense as it would help the model better learn to preserve content. Interestingly, feeding moral tokens to the decoder performed better than to the encoder, despite Chawla and Yang feeding moral tokens to the encoder. We think this is as the encoder, especially with pretraining, is better suited to preserving overall content, while the decoder is more important in the actual moral transfer process. Surprisingly, training the entire model did not yield the best performance. We believe this may have made the generator too complex and thus difficult to converge. We also noted that the Embedding Space content loss metric tended to improve performance, which may indicate that content is better represented in the embedding rather than latent space. This was further supported by observing outputs from models trained on Latent Space content loss, which had unrecognizable outputs. Our best model was pretrained, fed moral tokens to the decoder, had a frozen encoder, and used embedding-based content loss, and was sanity-checked by sampling outputs.

We then evaluated our best model on the full set of evaluation metrics. Given its strong validation performance, we are not surprised by its strong test performance when comparing the classifier predicted output to the input target morals. However, it is surprising that when using the NELA featurizer to classify our generated text in comparison to the input target morals, our best model performed poorly. This is interesting given that we did train our classifier on moral features obtained from the NELA featurizer. While the NELA featurizer is not perfect and cannot be considered a true ground truth, this may indicate our generator has overfit to our own classifier. The strong BERTSCORE, an external evaluation metric, is especially encouraging and indicates that the content loss metrics we developed were effective in training our model to preserve content while performing moral style transfer.

5 Analysis

5.0.1 Classifier

For our classifier, despite its high accuracy, we look at our incorrect classifications from the test set (see Table 5) to get a better sense of where our model may be failing. In some cases, we note that the ground truth created by NELA is itself wrong, such as examples 1 and 2, as it has both the virtue and vice of the same moral, which would be contradictory. This shows that NELA is itself not perfect and demonstrates our model’s ability to perform well in not predicting contradictory morals. Another common source of error is missing out on one or having one extra moral in a list of multiple morals, as seen in example 3 (one less moral) and 4 (one extra moral). In the case of having one less moral (usually a single prediction), this may indicate our model is learning to focus on only a single prediction and in the case of an extra moral this may indicate our model is thrown off by contextual

clues such as in example 4 with the word "nationalist." Finally, as in example 5, another source of error is our model picking up no morals. We believe this is as the NELA featurizer may be too sensitive in that it matches keywords and in this case is fooled by the word "virgin."

On the whole, while this error analysis presents steps for further improvement, we are not too concerned given the strong numeric performance of our model, and this may even indicate our model performs better than NELA when considering the errors.

#	Input	NELA original morals	Classifier predictions
1	Perspective One family's experience with the 'unfair' math of our country: Four U.S. citizens plus one who is not equals zero stimulus funds	Fairness-, Fairness+	Fairness-
2	Fascist Fauci, Whose Government-Guaranteed Paycheck is Secure, Emails NY Times Fake News Reporter With Warning of 'Needless Suffering and Death in Opening Country 'Prematurely'	Harm- Harm+	Harm+
3	Executive actions to extend unemployment benefits and defer payroll taxes signed by the President on Saturday brazenly skirt Congress	Authority+, Harm+	Harm+
4	Coronavirus shows Wales can cope alone, says nationalist leader calling for vote	Authority+	Authority+, Ingroup+
5	What's the difference between A.I. veterans and A.I. virgins?	Purity+	None

Table 5: Misclassified Results from Test Set.

Note that + denotes a morals' virtue and - denotes that moral's vice.

5.0.2 Generator

#	Input	Original Morals	Target Morals	Output
1	6 rules for running virtual meetings that will [...] make you a respected leader	Authority+	Fairness+	6 rules for running virtual meetings that will [...] make you a successful CEO
2	L.A. to add 6,000 shelter beds for homeless Angelenos in hopes of slowing virus' spread	Harm+	Harm- Ingroup+ Purity-	L.A. to add 6,000 emergency beds for homeless Angelenos in hopes of slowing virus' spread
3	Mary Trump to release new book on 'America's national trauma'	Ingroup+	Ingroup+ Purity+ Authority-	Mary Trump to release new book on 'America's regional trauma'
4	Mystery over absence of Algeria leader treated for COVID-19	Authority+	Authority- Purity-	Mystery over absence of Algeria chief treated for COVID-19
5	US judge cancels permit for Keystone XL pipeline from Canada	Authority+	Purity+ Harm+ Ingroup-	US judge cancels grant for Keystone XL pipeline from Canada

Table 6: Generated examples (met expectations).

Note that + denotes a morals' virtue and - denotes that moral's vice.

Overall, the qualitative generator results are very satisfactory. While preserving content, the generator modifies word(s) most closely correlated with the original moral into a state that resembles the target output. Sample 1 in Table 6, for example, identifies the authority-virtue associated with "leader" and disables this, transforming it to a synonym in "CEO" which may not necessarily be a good or bad form of authority. Similarly, the harm-virtue (which corresponds to care) in Sample 2 is transformed into a harm-vice in going from "shelter", a term indicative of a place where one may be cared and regarded for, to "emergency", which signifies some form of immediate harm.

Our generator is also able to effectively preserve certain morals while modifying other morals at the same time. In sample 3, the ingroup-virtue is preserved when transferring from "national trauma" to

"regional trauma", in both cases appealing to members of a geographically demarcated demographic - the United States. However, at the same time, the demotion from national to regional signifies a demotion in authority.

It is worth noting that specific words can correspond to multiple moral foundations. In sample 5, the generator changes "permit" to "grant" - at the same time, this demotes the authority associated with the former and associates the word "grant" with the care-virtue and ingroup-vice.

We also identified certain examples where the generator did not meet our expectations.

#	Input	Original Morals	Target Morals	Output
1	The coronavirus pandemic has radically redefined what it means to be clean	Purity+	Purity+ Harm− Ingroup−	The coronavirus pandemic has radically redefined what it means to be free
2	Foreign Office no-go warning to be lifted as England scraps quarantine requirement	Ingroup−	Ingroup+ Fairness+	Policy Office no-go warning to be lifted as England scraps quarantine requirement
3	New Database Documents Health Care Workers Who Died Fighting COVID-19	Harm−	Harm+ Authority+ Ingroup+	New Database Documents Health Life Workers Who Died Fighting COVID-19

Table 7: Generated examples (failed expectations).

Note that + denotes a morals' virtue and − denotes that moral's vice.

First, the generator's meaning preservation is not perfect. Sample 1 changes "clean" to "free" to incorporate the harm-vice and ingroup-vice, but this shifts the meaning of the original sentence from hygiene-related to a statement on freedom. Therefore, even though the morality transfer was correct, the meaning was altered and this does not align with our goal of retaining the original content.

Second, the model cannot discriminate proper nouns which are also transferred morally. For example, even though "Foreign Office" being changed to "Policy Office" makes sense in terms of going from ingroup-vice to ingroup-virtue (as the foreign office deals with outgroups) this loses the context of the input phrase, which refers to a real world entity. This is important because at least one of the moral foundations, authority, closely correlates with the use of proper nouns.

Finally, the generator also occasionally produces grammatical errors or nonsensical words. Sample 3 refers to health care workers, which is transformed into "health life workers". Although the meaning of the overall sentence remains easily inferrable, and the morality transfer worked quite well, the output is not contextually correct.

In general, the sample outputs do meet expectations. The generator makes powerful changes to the input sentences to subtly modify the moral presentation, which in turn may shift a piece of text to appeal to different audiences.

6 Conclusion

We developed a classifier that is extremely powerful in identifying the underlying morals in a text, and is even able to identify the underlying morals in the content of articles just based on the title. We have also developed a generator, the first of its kind, which is able to perform convincing moral style transfer. We learned the importance of preserving content in style transfer tasks and enjoyed developing and applying our own content preservation metrics suited for unsupervised style transfer tasks. We recognize the potential abuse of our work by bad actors, and thus would like to emphasize that its release be limited. That said, given the ongoing pandemic, we are excited to utilize this work to immediately help with framing COVID messaging and in the long term to help with reducing political polarization.

Our generator work was limited to news headlines, due to computational constraints, which may have simplified our task. Future work should explore performing moral style transfer at the document level, such as on entire news articles. In addition, our generated results should be evaluated in a psychology experiment to determine if they are effective in persuading people across different moral foundations.

References

- [1] Matthew Feinberg and Robb Willer. Moral reframing: A technique for effective and persuasive communication across political divides. *Social and Personality Psychology Compass*, 13, 12 2019.
- [2] Political polarization in the american public, June 2014.
- [3] J. Haidt and C. Joseph. The moral mind: How 5 sets of innate moral intuitions guide the development of many culture-specific virtues, and perhaps even modules. volume 3, 2007.
- [4] Ying Lin, Joe Hoover, Gwenyth Portillo-Wightman, Christina Park, Morteza Dehghani, and Heng Ji. Acquiring background knowledge to improve moral value prediction. In *2018 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*, pages 552–559. IEEE, 2018.
- [5] Benjamin D. Horne, Jeppe Nørregaard, and Sibel Adali. Robust fake news detection over time and attack. *ACM Trans. Intell. Syst. Technol.*, 11(1), December 2019.
- [6] Oscar Araque, Lorenzo Gatti, and Kyriaki Kalimeri. Moralstrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction. *Knowledge-based systems*, 191:105184, 2020.
- [7] K. Chawla and D. Yang. Semi-supervised formality style transfer using language model discriminator and mutual information maximization. 2020.
- [8] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.
- [9] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface’s transformers: State-of-the-art natural language processing, 2020.
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [11] Ruochen Xu, Tao Ge, and Furu Wei. Formality style transfer with hybrid textual annotations. *arXiv preprint arXiv:1903.06353*, 2019.
- [12] Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. Sentence mover’s similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760, Florence, Italy, July 2019. Association for Computational Linguistics.
- [13] Benjamin Horne and Mauricio Gruppi. NELA-GT-2020, 2021.
- [14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [15] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675, 2019.
- [16] WA Falcon and .al. Pytorch lightning. *GitHub. Note:* <https://github.com/PyTorchLightning/pytorch-lightning>, 3, 2019.

A Appendix (optional)

Pretraining	Input Strategy	Frozen Model	Content Loss Metric	Accuracy	Micro	Macro
Pretraining	Feed Encoder	Frozen Decoder	Latent Space	0.008	0.000	0.000
Pretraining	Feed Encoder	Frozen Decoder	Embedding Space	*	*	*
Pretraining	Feed Encoder	Frozen Neither	Latent Space	*	*	*
Pretraining	Feed Encoder	Frozen Neither	Embedding Space	0.008	0.000	0.000
Pretraining	Feed Decoder	Frozen Encoder	Latent Space	0.006	0.035	0.028
Pretraining	Feed Decoder	Frozen Encoder	Embedding Space	0.989	0.998	0.998
Pretraining	Feed Decoder	Frozen Neither	Latent Space	*	*	*
Pretraining	Feed Decoder	Frozen Neither	Embedding Space	*	*	*
Pretraining	Contextual Injection	Frozen Encoder	Latent Space	0.008	0.110	0.085
Pretraining	Contextual Injection	Frozen Encoder	Embedding Space	*	*	*
Pretraining	Contextual Injection	Frozen Decoder	Latent Space	*	*	*
Pretraining	Contextual Injection	Frozen Decoder	Embedding Space	*	*	*
Pretraining	Contextual Injection	Frozen Neither	Latent Space	*	*	*
Pretraining	Contextual Injection	Frozen Neither	Embedding Space	*	*	*
No Pretraining	Feed Encoder	Frozen Decoder	Latent Space	0.007	0.000	0.000
No Pretraining	Feed Encoder	Frozen Decoder	Embedding Space	*	*	*
No Pretraining	Feed Encoder	Frozen Neither	Latent Space	0.012	0.000	0.000
No Pretraining	Feed Encoder	Frozen Neither	Embedding Space	*	*	*
No Pretraining	Feed Decoder	Frozen Encoder	Latent Space	0.982	0.997	0.997
No Pretraining	Feed Decoder	Frozen Encoder	Embedding Space	*	*	*
No Pretraining	Feed Decoder	Frozen Neither	Latent Space	*	*	*
No Pretraining	Feed Decoder	Frozen Neither	Embedding Space	*	*	*
No Pretraining	Contextual Injection	Frozen Encoder	Latent Space	0.010	0.000	0.000
No Pretraining	Contextual Injection	Frozen Encoder	Embedding Space	*	*	*
No Pretraining	Contextual Injection	Frozen Decoder	Latent Space	*	*	*
No Pretraining	Contextual Injection	Frozen Decoder	Embedding Space	*	*	*
No Pretraining	Contextual Injection	Frozen Neither	Latent Space	*	*	*
No Pretraining	Contextual Injection	Frozen Neither	Embedding Space	*	*	*

Table 8: Moral Reframer validation results (all values are F1 scores unless noted). Note that any * results signify early stopping due to training stagnation, and these metrics do not reflect content consistency.

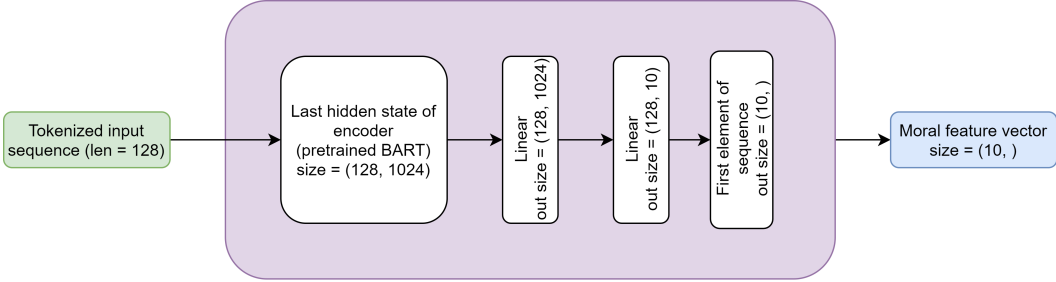


Figure 1: Diagram of the classifier architecture.

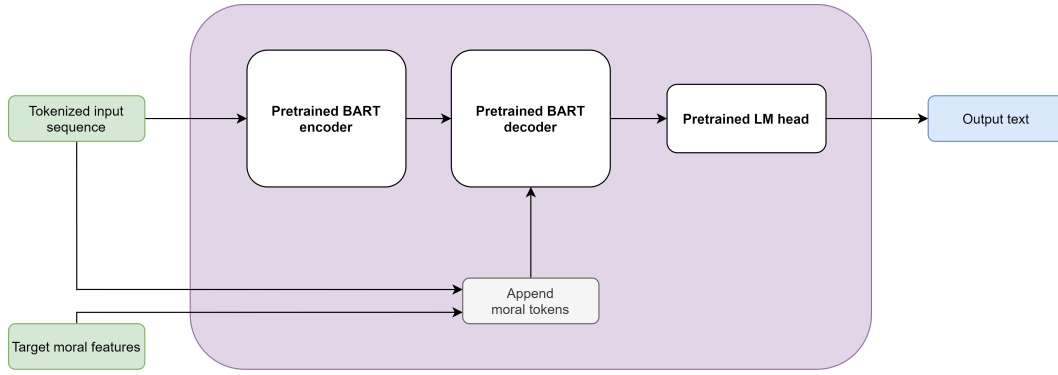


Figure 2: Diagram of the decoder architecture, with target morals as part of the decoder input sequence.

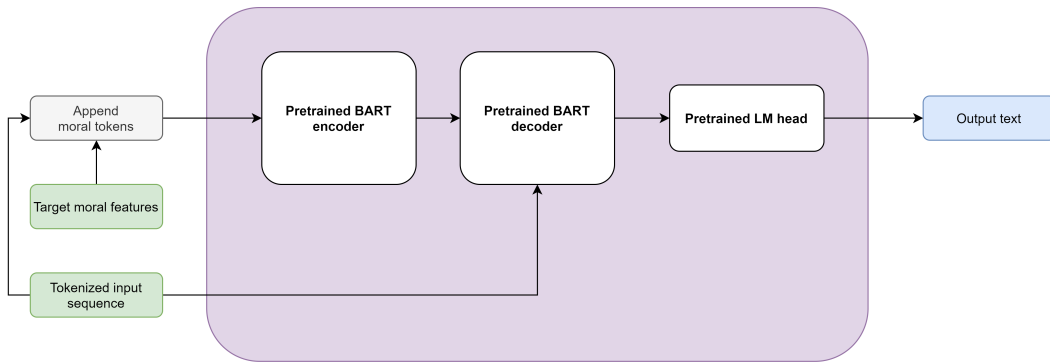


Figure 3: Diagram of the decoder architecture, with target morals as part of the encoder input sequence.

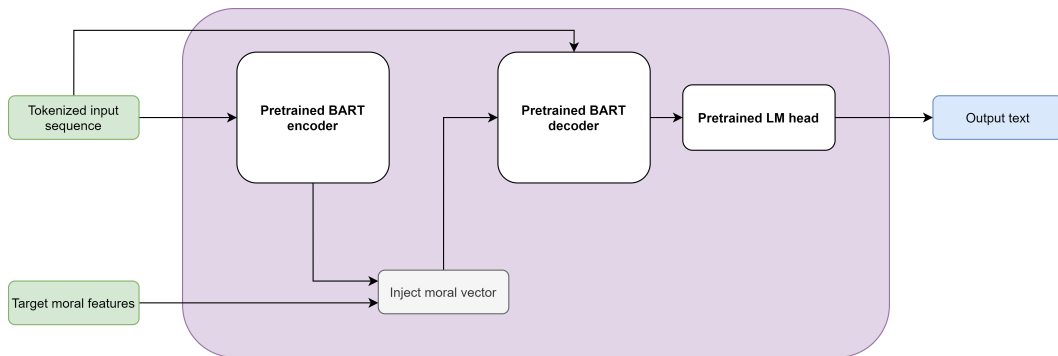


Figure 4: Diagram of the decoder architecture, with target morals injected directly into the latent space.

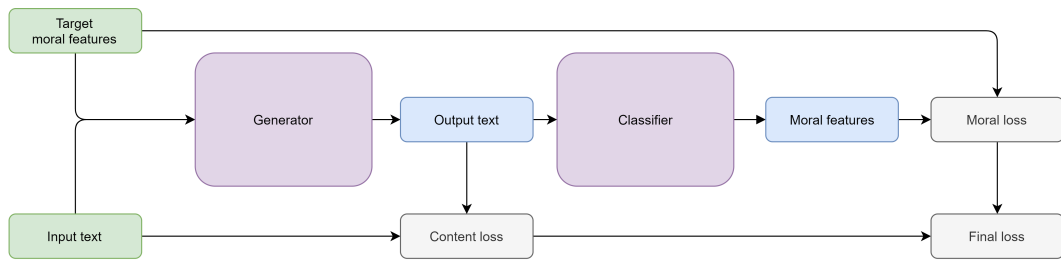


Figure 5: Diagram of the generator during training.