Sentiment Analysis & Text Style Transfer on Semi-Parallel Movie Critic Review Corpora Using NLP Translation Techniques

Charlie Glass
August 2023

Abstract

The recent dramatic increase in public consumption generative AI tools—like Chat GPT—has created the possibility of almost instantaneously writing original content in the styles of different authors—even those who have long passed away. This research focuses on such text-style transfer trained on semi-parallel corpora, using Encoder-Decoder and Sequence-to-Sequence architectures for two tasks: translating a positive review into a negative review, and translating reviews of the late, popular movie critic Roger Ebert into generic critic reviews. We also trained sentiment analysis models (using multilayer perceptron neural networks, CNNs, BERT pretrained models with CNNs), and used the top-performing sentiment analysis model to evaluate the success of the positive-negative translation capturing the right sentiment. The results from our research show significant performance boosts over baseline for sentiment analysis, but little to no improvement over simple baseline translations for our two text style transfer projects in relevance and fluidity.

Keywords: Text style transfer, semi-parallel corpora, sentiment analysis, relevance

Introduction

The goal of this research was to observe the effectiveness of Encoder/Decoder architectures used for language translation for different style translations: can Sequence-to-Sequence models translate movie reviews from the “language of positive” to the “language of negative”? Can they translate Roger Ebert’s review language to other critics’ reviews? Roger Ebert is perhaps the most famous movie critic in American film history and was known for his unique style and “rule breaking”, such as beginning reviews with long sentences or dialogue or subordinate clauses.[1] Such writing can be difficult for those without a film background, without a specialty English education, or with a different primary language than English to understand. We can see the public using models like what we employed to extract meaning from potentially complicated text into something more easily understandable. We can also envision this progressing

[1] McMahon, Jeff, “Why Roger Ebert Was The Greatest Movie Reviewer”, 2013, <https://www.forbes.com/sites/jeffmcmahon/2013/04/08/why-roger-ebert-was-the-greatest-movie-reviewer/?sh=617ef8427aa9>

further by reversing translation, keeping Ebert's voice, writing, and potentially opinion alive on new films despite his passing over ten years ago[2].

Furthermore, if the hypothesis that popular translation models could generate fluid, relevant text style transfers were deemed correct, popular translation tools like Google Translate could potentially incorporate simple text style transfer more quickly. Chat GPT has shown success with text style transfer[3], but not without relatively significant boundaries of entry, namely an OpenAI account and longer, possibly few-shot prompts that not everyone knows or knows how to write. With our proposed translation models, tools like Google Translate could quickly incorporate style transfers of any two styles and, provided enough semi-parallel training corpora, quickly process short blurbs into the translated style with existing translation network architectures.

Background

Sentiment analysis on movie reviews was popularized by review aggregator sites like Rotten Tomatoes and Metacritic, and has since been a popular deep learning exercise. The sentiment model we designed largely builds off of 2021 literature by researchers Ping Huang, Hujuan Zhu, Lei Zheng, and Ying Wang, who used BERT pre-trained embeddings and CNNs[4].

Though our project was limited by computer resources, our work is inspired by many text-style transfer research projects. Our main inspirations came from research conducted by Martina Toshevska and Sonja Gievska of Cyril and Methodius University, who described using an Encoder-Decoder architecture on parallel and semi-parallel corpora. However, our sentence pairs for both translation projects were often trained on much less parallel corpora than their examples. For example, one training sentence pair on the same movie discusses very different topics: *"After "Thor," this makes Marvel Comics two-for-two so far this summer movie season*" paired with the negative input sentence *"Imagine how disappointed I am to find this movie to be a scattered affair that loses power the more it tries to take on big issues*". This highlights one of the main challenges our models face: though the model is trained on pairs of reviews from the same movie, they could be positive or negative for totally separate reasons. The training sentences can also be very culturally specific, with proper nouns like *"Marvel Comics"* and American phrases like "*two-for-two*". We hoped to implement architectures

[2] Martin, Douglas, "Roger Ebert Dies at 70; a Critic for the Common Man", 2013, <https://www.nytimes.com/2013/04/05/movies/roger-ebert-film-critic-dies.html>
[3] Lai, Huiyuan, "Multidimensional Evaluation for Text Style Transfer Using ChatGPT", 2023, <https://arxiv.org/abs/2304.13462>
[4] Huang, Ping, Zhu, Huijuan, Zheng, Lei, Wang, Ying, "Text Sentiment Analysis based on BERT and Convolutional Neural Networks", 2021, <https://dl.acm.org/doi/fullHtml/10.1145/3508230.3508231>

proposed in research by Zhu et al.[5] and Shen et. Al[6], which discusses using a Cross-Aligned Auto-Encoder featuring an Encoder and Generator model for transfer on non-parallel corpora, but our limited RAM and GPU crashed after implementation.

Approach

Data

All the data for this project comes from a Rotten Tomatoes Critic Reviews Dataset from Kaggle[7], which totaled approximately 1.1M rows. Each row is an individual row, with fields *rotten_tomatoes_link* (the movie being reviewed), *critic_name*, *review_type* (review was deemed "Fresh" [positive] or "Rotten" [negative] by Rotten Tomatoes), *review_score* (optional field with the critic's own numeric or letter grade), and *review_content* (a short headline summing up the review). We immediately dropped any rows with null values for movie, critic name, review type, or review content. We also found several duplicate reviews with the same movie and critic, and some reviews that were in languages other than English. These rows were also dropped, leaving around 900K total rows. For our sentiment analysis model, we used the review headline sentences as inputs with the binary "Fresh" or "Rotten" review type as the target. For our sentiment transfer and Ebert-to-generic transfer models, we used negative/positive and Ebert/other critic sentence pairs as inputs. For the Ebert model, we trained on multiple sentence pairs for a given Roger Ebert/movie review combination (i.e. the movie *In Her Shoes* featured the same review Ebert did for it, joined onto reviews that other critics did on it). For the sentiment transfer model, we tried various different sentence pairings: One sentence pair with a given movie, critic name, positive review combination, one sentence pair for a given movie, one sentence pair with a given movie, critic name, positive review combination, deduped on closest review length matches, and one sentence pair with a given movie, critic name, positive review combination, deduped randomly on reviews less than 85 characters in length. For these models, the number of total pairs ranged between 300K and 2M. For every model, the data were split randomly into 80%/10%/10% training/validation/test splits.

Baseline

The baseline model for our sentiment analysis model was simply always predicting positive sentiment, since 63% of the training data were positive. For a more complex baseline, we ran simple Neural Networks with Bag-of-Words inputs. Our sentiment transfer model also had a simple baseline, predicting the negative sentences "*the film was terrible*" and "*the worst movie of the year*" at random. The Ebert transfer model was similar, predicting "*the film was great*" for positive Ebert reviews or "*the film was terrible*" for negative Ebert reviews.

---

[5] Zhu, Xuekai, Guan, Jian, Huang, Minlie, Liu, Juan. "StoryTrans: Non-Parallel Story Author-Style Transfer with Discourse Representations and Content Enhancing, 2023, <https://arxiv.org/pdf/2208.13423.pdf>

[6] Shen, Tianxiao, Lei, Tao, Barzilay, Regina, Jaakkola, Tommi, "Style Transfer from Non-Parallel Text by Cross-Alignment", 2017, <https://arxiv.org/pdf/1705.09655.pdf>

[7] <https://www.kaggle.com/datasets/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset?resource=download>

Modeling
For sentiment analysis, we used various Neural Network classifier architectures, including WANs, DANs, CNNs (all with BERT embeddings), as well as a BERT pretrained model with CNN layers. For each of our transfer models, we ran Sequence-to-Sequence models with Keras layers for token and position embedding, an encoder, and a decoder to predict the maximum likelihood sentence in the given style. Along with variance in input sentence pairs (previously described in the *Data* section), we tested different original and styled vocabulary sizes and number of epochs. We alternated between 30,000 and 100,000 vocabulary sizes. The number of epochs ranged between 15 and 40. We kept all other major specifications the same for transfer modeling (see Appendix for specifications list). We attempted to model with BERT pre-trained embeddings and model, and the Cross-Alignment Auto-Encoder architecture, but our system's RAM and GPU crashed with both implementations.

Evaluation
Our primary metric for sentiment analysis was accuracy. For additional verification, we looked at the distribution of critic-filled review scores for correct and incorrect predictions.

Our transfer models were trickier to evaluate. For sentiment transfer, we used our top performing sentiment analysis model to see how often the translated sentence was negative. Our main focuses of transfer style evaluation were if the translation was about a movie, if the sentiment matched the input's, if the translation contained specific references that the input referenced (i.e. the cast, the specific movie, the movie length, etc.), and fluidity. We believed that existing metrics, like BLEURT, failed to capture all of these, and current implementations of them were better suited for language translation than for style transfer: we care more about a fluid, relevant translation than word overlap. Therefore, we chose to incorporate human evaluators for each of the four topics, asking them to fill binary responses for the 3 binary topics and rate the fluidity 1-10. The human judges scored 4963 samples, and scores were normalized to reflect potential differences among the judges' scoring preferences. The translations included the baseline translations, and translations were sorted and blinded as to which model the translation came from. For future research, we would like to test out using BLEURT for evaluation.

Results

Our pretrained BERT model with CNN layers showed significant performance boosts above our baseline. The test accuracy of our always-positive prediction baseline model was 64%, and the bag-of-words models had a test accuracy of ~70%. Our BERT CNN model produced a test accuracy of 89% (see Appendix IV for full results), with most of the incorrect predictions coming on middling review scores:

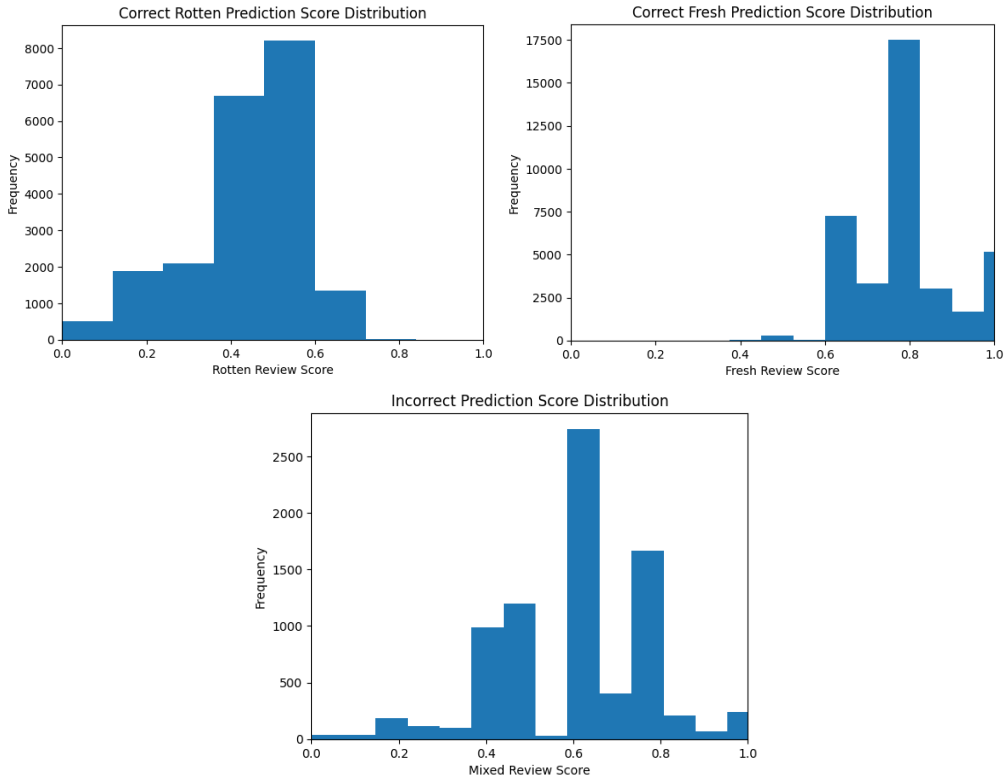


Figure 1: On a 0-1 scale, most incorrect predictions came between 0.4 and 0.6. Note that letter grades A,B,C,D,F were translated to 1, 0.8, 0.6, 0.4, and 0.2, resulting in some bars sticking out.

The sentiment transfer model was excellent at resulting in negative translations. As evaluated by the sentiment analysis model, a sample of 500 translations were negative between 84% and 95% of the time. Human judges also found the translation to be about a movie 100% of the time, as each translated sentence started with "*the film*", "*the film's*", "*the movie*", or "*the movie's*". However, the sentiment transfer model showed little to no improvement over baseline for its relevance score and fluidity score. The baseline contained only 4% of translations with relevant, specific information from the original, whereas the Sequence-to-Sequence models ranged between 0%-14%[8]. Based on the confidence interval of each, these scores were not statistically significant improvements over the baseline. Fluidity scores were mostly worse than the baseline's fluidity score with 95% confidence. The mean fluidity score for the baseline was 7.06, whereas all but one sentiment transfer model resulted in mean scores of 2.97-4.18. Models run on 40 epochs had the highest relevance scores, but still not confidently above the baseline. The models often outputted awkward, unusable sentences as one example illustrates: *"the film ' s biggest problem is that the film is , in the end , and the film ' s lack of a story that is so much more than a a film that is a bit too much , and the film is so predictable".* Our transfer model trained on shorter sequences of <85 characters with duplicate critic/movie/positive review combinations had the highest mean fluidity score of 9.38, but with a standard deviation of 1.94, the model failed to perform above the baseline with statistical significance. The model only produced 4 unique sentences, with one sentence produced the vast majority of the time that always

[8] See Appendix I for full results

outputted >95th percentile fluidity scores: "*the film's second half, however, disappoints in many respects."* This resulted in a tradeoff with relevance, with only 2% of the model's translations coming in as relevant.

The Ebert transfer model suffered from similar issues. While the sentiment matched 75% of the time and all translations were recognized as being about a movie, the translations had a mean fluidity score of 3.3, compared with 7.43 for the baseline. The translations could also be awkward: *"a film that is not only a great film , but also a little too long and too long and too often a little too long , and too much of the film is a little too much*". The relevance score mean was 0.14 compared to 0.05 for the baseline, but with a standard deviation of 0.35, the relevance score was not significant. Sentences with multiple training pairs all discussing the same topic tended to have higher relevance scores: "*the wrestler is a film about the human spirit, and the best of the year*". Below we see full scores for the Ebert transfer model:

Model	**Sentiment matches (mean, std)**	**About a movie (mean, std)**	**Contains relevant info (mean, std)**	**Fluidity Score 1-10 (mean, std)**
Baseline	(0.96, 0.07)	(1.0, 0.0)	(0.05, 0.09)	(7.43, 0.78)
Ebert Seq2Seq Transfer Model	(0.76, 0.43)	(1.0, 0.0)	(0.14, 0.35)	(3.33, 2.02)

Conclusions & Next Steps

While our sentiment analysis model performed well, our style transfer models did not perform well enough over baseline (particularly in relevance and fluidity) to recommend using standard translation Sequence-to-Sequence models for text style transfer on semi/non-parallel corpora. Our transfer models with more epochs and more training pair examples showed more promise in relevance and fluidity. Intuitively, this makes sense: since there are multiple ways to express a critic review negatively or generically on non-parallel corpora, a model that generates these text style transfers should require even more sentence pairs—expressing the same input sentence differently—than language translation models. For future work, we recommend collecting more sentence pairs for further training, as well as possibly expanding from review headlines to full review articles for even more training data.

The models also clearly suffered from their embeddings. Since we used Keras to define our own embeddings based on just the vocabulary in our dataset, the translations included only the most common words the vast majority of the time (film, movie, well-made, good, and bad were overwhelmingly the most popular words in the translations, with other words seldom appearing). We would like to explore using Huggingface's pre-trained BERT Encoder-Decoder model and embeddings for a better understanding of context and expanded vocabularies. We also hope to explore using the Cross-Alignment Auto-Encoder architecture proposed by Shen et. al to test out a more sophisticated, tested solution to text style transfer on non-parallel corpora, along with using a beamwidth search of >1 for multiple potential sentence outputs.

Full References

1. McMahon, Jeff, “Why Roger Ebert Was The Greatest Movie Reviewer”, 2013, https://www.forbes.com/sites/jeffmcmahon/2013/04/08/why-roger-ebert-was-the-greatest-movie-reviewer/?sh=617ef8427aa9
2. Martin, Douglas, “Roger Ebert Dies at 70; a Critic for the Common Man”, 2013, https://www.nytimes.com/2013/04/05/movies/roger-ebert-film-critic-dies.html
3. Lai, Huiyuan, “Multidimensional Evaluation for Text Style Transfer Using ChatGPT”, 2023, https://arxiv.org/abs/2304.13462
4. Huang, Ping, Zhu, Huijuan, Zheng, Lei, Wang, Ying, “Text Sentiment Analysis based on BERT and Convolutional Neural Networks”, 2021, https://dl.acm.org/doi/fullHtml/10.1145/3508230.3508231
5. Zhu, Xuekai, Guan, Jian, Huang, Minlie, Liu, Juan. “StoryTrans: Non-Parallel Story Author-Style Transfer with Discourse Representations and Content Enhancing, 2023, https://arxiv.org/pdf/2208.13423.pdf
6. Shen, Tianxiao, Lei, Tao, Barzilay, Regina, Jaakkola, Tommi, “Style Transfer from Non-Parallel Text by Cross-Alignment”, 2017, https://arxiv.org/pdf/1705.09655.pdf
7. <https://www.kaggle.com/datasets/stefanoleone992/rotten-tomatoes-movies-and-critic-reviews-dataset?resource=download>

Appendix I: Sentiment Transfer Model Results

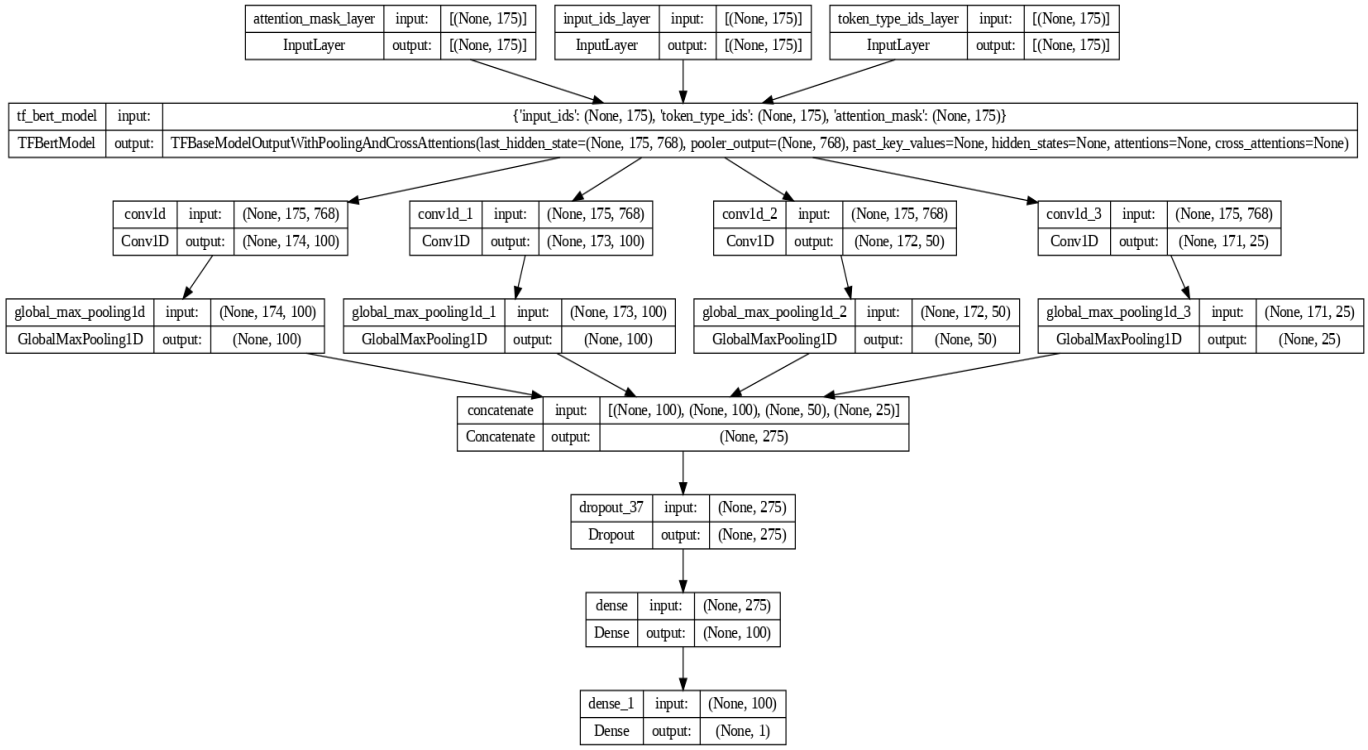
Model	**About a movie (mean, std)**	**Contains relevant info (mean, std)**	**Fluidity Score 1-10 (mean, std)**
Baseline	(1.00 , 0.00)	(0.04, 0.20)	(7.06, 0.43)
Negative Sentiment Transfer (20 Epochs)	(1.00 , 0.00)	(0.05, 0.23)	(3.50, 2.02)
Negative Sentiment Transfer (40 Epochs)	(1.00 , 0.00)	(0.14, 0.36)	(4.18, 2.16)
Negative Sentiment Transfer (<85 characters, repeat pairs, 10 epochs)	(1.00 , 0.00)	(0.02, 0.15)	(9.38, 1.94)
Negative Sentiment Transfer (matched on close lengths, 40 epochs)	(1.00 , 0.00)	(0.06, 0.25)	(2.97, 2.11)
Negative Sentiment Transfer (matched on	(1.00 , 0.00)	(0.00, 0.00)	(2.97, 1.72)

close lengths, 25 epochs)			
Negative Sentiment Transfer (100,000 Vocab Size, 25 epochs)	(1.00 , 0.00)	(0.00, 0.00)	(3.04, 1.00)

Appendix II: Standard Transfer Model Specifications

Batch Size: 64
Epochs: 10, 20, 25, 40
Max Sequence Length: 125
Original Vocabulary Size: 30,000/100,000
Styled Vocabulary Size: 30,000/100,000
Embedding Dimensions: 256
Intermediate Dimensions: 2048
Attention Heads: 8

Appendix III: Sentiment Analysis BERT-CNN Model Structure



Appendix IV: Sentiment Analysis Classification Model Results

Model Description	Test Accuracy
Baseline	64%
Bag-of-words Logistic Regression, 500 word vocab	70%
Bag-of-words Neural Network, size-15 hidden layer, 500 word vocab	70%
Bag-of-words Neural Network, size-15 hidden layer, 1000 word vocab	73%
DAN model BERT embeddings	74%
CNN model BERT embeddings	81%
BERT pre-trained model with CNN	88%

Appendix V: Sample Seq2Seq Style Transfer Model Structure
Model: "s2sTransformer"

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
==================================================================
 encoder_inputs (InputLayer  [(None, None)]            0         []
 )

 token_and_position_embeddi  (None, None, 256)         7712000
['encoder_inputs[0][0]']
 ng (TokenAndPositionEmbedd
 ing)

 decoder_inputs (InputLayer  [(None, None)]            0         []
 )

 transformer_encoder (Trans  (None, None, 256)         1315072
['token_and_position_embedding
 formerEncoder)                                        [0][0]']

 model_2 (Functional)        (None, None, 30000)       1700075
['decoder_inputs[0][0]',
                                                        2         'transformer_encoder[0][0]']

==================================================================
Total params: 26027824 (99.29 MB)
Trainable params: 26027824 (99.29 MB)
Non-trainable params: 0 (0.00 Byte)
```