Self-supervised text de-stylization based on BERT

Junyang Huang[1,3], **Xiaoxiao Lin**[2]

[1]Software Engineering, LUT University, Mukkulankatu 19, 15210 Lahti, Finland
[2]Computer Science, University of Bristol, Bristol, BS8 1QU, UK

[3]Junyang.Huang@student.lut.fi

Abstract. Recent advancements in Natural Language Processing (NLP) have ushered in a new era of textual style transfer (TST), a domain aimed at altering textual attributes such as tone and sentiment while preserving the content's essence. This study introduces a creative framework that employs a dual-component architecture consisting of a classifier and a generator to achieve text de-stylization, particularly sentiment neutralization. The classifier, built upon the Bidirectional Encoder Representations from Transformers (BERT) model, serves as a dynamic loss function guiding the generator, constructed on a Transformer-based encoder-decoder framework, to produce sentiment-neutral text. Our method leverages a self-supervised mechanism, enabling the generation of target text without reliance on parallel corpora, thereby addressing the limitations of existing TST methodologies. We preprocessed datasets from Stanford Sentiment Treebank-5 (SST-5) and Internet Movie Database (IMDb) movie reviews and employed them for training the classifier and generator, respectively. Preliminary results demonstrate the model's proficiency in preserving semantic integrity while effectively neutralizing sentiment. Future work envisions expanding this framework to enable text stylization across a spectrum of discursive contexts, enhanced by deep learning architectures and an iterative feedback mechanism for user-driven refinement.

Keywords: Text De-stylization, seif-supervised learning, Transformer

1. Introduction

Textual style transfer (TST) has been a popular topic in the realm of Natural Language Processing (NLP) in recent years. The primary objective of TST is to modify the stylistic attributes of a text, such as its tone, sentiment, or formality, while preserving its original content and meaning. This domain has its roots in the successful application of Generative Adversarial Networks (GANs) to image style transfer [1], however, poses unique challenges not present in visual parts. Unlike images, where style can be represented as patterns, colors, and textures, the style in text encapsulates a large variety of attributes including lexical choice, syntactic structures, emotion, intent, and even socio-cultural variances like gender or educational background. Consequently, analyzing text styles is a complex process, which has been the most key step of the textual style transfer task.

Previous TST works can mainly be divided into the supervised-based and unsupervised methods. The supervised-based method directly model the sequence-to-sequence relation from original domain to target domain for text style transfer. For instance, Jhamtani et al. [2] implemented an automated method for converting modern English into Shakespeare's English by constructing an external dictionary. However, preparing distinct datasets [3] for varying style transfer tasks is labor-intensive, which restricts the task to an unsupervised framework. Unsupervised-based method especially gives the independence from parallel corpora, which can be broadly dichotomized into: (1) **Explicit Strategies**. These involve external manipulations, such as the removal or replacement of specific emotional words, adjectives, or words with evident tendencies. The underlying assumption is that certain lexemes inherently carry stylistic weight, and their substitution can effectuate a change in the style of the text. (2) **Implicit Strategies**. Implicit methods aim to identify and separate latent representations of textual content and its associated style in the latent space. This strategy is also called disentanglement, allowing for distinct encodings of style and content. For instance, adversarial learning facilitates the creation of distinct latent spaces for content and style, ensuring minimal overlap. Another approach involves the guided editing of latent representations under the aegis of attribute classifiers. Moreover, some methods adopt a dual latent representation framework, encoding the input text separately for attributes and content, facilitating style transfer by interchanging the attribute representations. Considering the lack of parallel corpus, separating text style and content is the mainstream strategy in the field of text style transfer, though a considerable variety of branches have been developed under this. However, this strategy still inevitably brings some problems, especially the issue of content loss. The prevalent practice of disentangling text into style and content components may inadvertently compromise textual coherence and result in loss of integral content.

To alleviate the aforementioned challenge, we propose to convert styled text into a neutral, attribute free version based on the Bidirectional Encoder Representations from Transformers (BERT), so that provides a basic foundation for subsequent operations. To execute style transfer without compromising the text's core content, we have designed four technical steps to implement the restoration process and avoid loss of text content:

(1) Sentiment classifier: The task of a sentiment classifier is to identify and label the emotional attributes of text (such as positive or negative). This information provides important clues about the emotional characteristics that need to be neutralized.

(2) Encoder-Decoder Framework: We will use this structure to encode text and attempt to produce an emotionally neutral version. In this way, the style features of the text are "removed" while the core content is preserved.

(3) Self-Supervised mechanism: In the process of de stylization, sentiment classifiers are not only used to preliminarily determine the emotions of the text but can also serve as self supervised mechanism. During the generation process, the classifier can provide immediate feedback to guide the decoder in adjusting towards producing emotional neutral text. In this way, we use known emotional labels to supervise and guide the stylization process, ensuring that the output results are consistent with expectations.

(4) Content discriminator/attention mechanism: To ensure the consistency of text content, we will introduce a content discriminator. The task of this discriminator is to ensure that the stylized text remains semantically consistent with the original content. In addition, attention mechanism is a good candidate, which can make the model pay more attention to the content of the original text during the generation process.

2. Revisiting BERT

The rapid evolution of NLP owes much to architectures like the Transformer, which paved the way for models of Bidirectional Encoder Representations from Transformers (BERT). BERT is the inaugural representation model that utilizes fine-tuning to attain unparalleled results across an extensive array of sentence-based and token-based tasks, surpassing numerous specialized structures [4]. BERT's bidirectional context and self-supervised training methodology, particularly its masked language modeling, make it adept at text destylization task. The BERT model has the following characteristics:

(1) **Bi-directional context representation**: Although models such as Generative Pre-trained Transformer (GPT-2) also use the Transformer structure, each output can only focus on the previous output. Unlike previous models, BERT can simultaneously consider the left and right context of words, generating richer and more accurate word embedding representations.

(2) **Pre-training and fine-tuning**: The BERT model is divided into two stages: pre-training and fine-tuning. During the pre-training stage, the model is trained on a large amount of unlabeled text to learn the basic features of the language. In the fine-tuning stage, the model is adjusted and trained on specific NLP tasks to adapt to different needs.

(3) **Pre-training task**: BERT uses two different tasks for pre-training: Masked Language Model (MLM) and Next Sentence Prediction (NSP). MLM: Randomly mask some words in a sentence and train the model to predict these masked words. NSP: Train a model to predict whether one sentence is the next sentence of another.

(4) **Model architecture**: BERT uses the Transformer architecture, including a multi-layer bidirectional Transformer encoder. Two versions are provided: BERT-BASE (12 layers, 768 hidden units, 12 attention heads) and BERT-LARGE (24 layers, 1024 hidden units, 16 attention heads).

(5) **Multi-task adaptability**: BERT can easily adapt to different NLP tasks with minimal structural changes.Simply by replacing the input and output layers, it can be used for tasks such as text classification, sequence labeling, and question answering.

BERT processes either a solitary sentence or a duo of sentences (e.g., the combination of 'question' and 'answer') into a series of tokens, leveraging WordPiece embeddings [5]. Every sequence starts with the "[CLS]" token. If there's a sentence pair, they are demarcated by the “[SEP]” token. Additionally, each token is enhanced with an embedding that denotes if it's part of the initial or subsequent sentence. The overall representation for a specific token is derived by amalgamating its respective token, position, and segment embeddings.

3. Transformer for De-stylization

3.1. Text De-stylization

The task of text de-stylization aims at transforming the sentiment attribute of a given textual content from a polarized emotion (positive or negative) to a neutral tone, while preserving the core semantic information. Mathematically, this can be formulated as follows:

Given an input text sequence $X = \{x_1, x_2, \ldots x_n\}$ associated with the sentiment label $S_X \in 0, 2$, where 0 and 2 represent negative and positive sentiments respectively, the goal is to generate a corresponding neutral text sequence $Y = \{y_1, y, \ldots y_m\}$ such that its sentiment label $S_Y = 1$, indicating a neutral label. It's crucial to ensure that the semantic essence of the original text is retained in the transformed output, implying that the information content $I(X)$ should be approximately equal to $I(Y)$, where $I(\cdot)$ denotes the information content of the text.

In this context, the sentiment attribute transformation can be visualized as a mapping function $f: X \times S_X \rightarrow Y \times S_Y$, which not only converts the sentiment attribute but also ensures the fidelity of the content. This process involves the identification and substitution of sentiment-bearing words while maintaining the contextual relevance derived from the non-sentiment-bearing components of the text [6].

The complexity of this task stems from the intricate interplay between sentiment-bearing and non-sentiment-bearing elements within the text. It requires a nuanced understanding of the contextual implications of each word and the overall sentiment conveyed by the text as a whole. Therefore, the objective of our model is not merely to replace words with their neutral counterparts but to comprehensively restructure the sentiment attribute while preserving the original content's integrity.

3.2. Model Overview

As shown in Figure 1, the propose method for text de-stylization incorporates a dual-component architecture consisting of a classifier and a generator. The classifier is built upon the BERT model, while the generator is constructed on a Transformer-based encoder-decoder framework. Each serves a distinct yet complementary role in the de-stylization process.

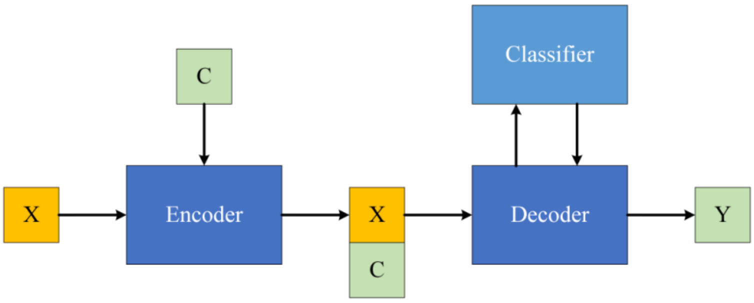

Figure 1. Overview of the propose method for text de-stylization

3.3. Classifier

The classifier, built upon the robust BERT model, serves as a sentiment analysis and supervised tool. It evaluates the sentiment of the text generated by the generator, providing feedback that informs the subsequent iterations of text generation. This creates a self-supervised loop, with the classifier acting as a dynamic loss function. The model aims to minimize the divergence between the classifier's sentiment prediction and the neutral sentiment target. Mathematically, the classifier's role can be represented as:

$$Loss = D(Classifier(Y), NeutralSentiment) \tag{1}$$

where$D$ denotes cross-entropy loss, and $NeutralSentiment$ represents the neutral sentiment target.

3.3.1. Classifier architecture. The architecture of classifier is structured as follows: (1) **BERT Model:** A pre-trained BERT processes input text into a sequence of token embeddings. Each token is represented by a vector in a high-dimensional space. The model is pre-trained on large corpora, enabling it to understand complex language patterns. For classification tasks, the embedding of the first token, often the <CLS> token, is used as the aggregate sequence representation. (2) **Dropout Layer:** This layer randomly nullifies a subset of its inputs (denoted as p) during training, preventing overfitting and promoting generalization. It can be represented as $r \odot x$, where r is a mask vector with elements drawn from a Bernoulli distribution with probability p , and x is the input vector. (3) **Linear Layer:** A linear transformation that maps the 768-dimensional pooled output from the BERT model to a 3-dimensional space corresponding to our sentiment classes. If x is the input vector and W and b are the weight matrix and bias vector respectively, the transformation is $Wx + b$.

3.3.2. Forward Pass. During the forward pass, the input text is tokenized and converted into a sequence of input IDs. Alongside, attention masks are created to differentiate actual content from padding. The BERT model takes these IDs and masks, and outputs the pooled representation of the input sequence. This representation, denoted as h , undergoes the dropout operation $r \odot h$, followed by the linear transformation $W(r \odot h\ ) + b$, yielding the classification logits.

3.3.3. Classification Loss. The training process involves minimizing the CrossEntropy loss, a measure of the discrepancy between predicted sentiment probabilities and true labels [7]. If y is the true label vector and o is the output logits, the CrossEntropy loss is computed as $-\Sigma y \log(soft\max(o))$.

3.4. Generator

We have built an encoder decoder structure based on the Transformer framework to achieve text generation. Due to the introduction of self supervised mechanism, the generator can generate target sentiment text without relying on the target sequence.

The generator's primary function is to produce neutral-toned text from sentiment-laden input. It employs a Transformer-based architecture renowned for its effectiveness in sequence-to-sequence tasks.

A novel feature of our generator is the integration of control codes within the encoder. These codes, representing neutral sentiment, guide the model in generating text devoid of emotional bias. Mathematically, the encoder maps an input sequence X to a latent representation Z, which is then transformed by the decoder into a neutral sentiment sequence Y, as follows:

$$Z = Encoder\big(Concat(X,\ ControlCode)\big) \tag{2}$$

$$Y = Decoder(Z) \tag{3}$$

The workflow begins with the generator receiving a sentiment-laden text and a neutral sentiment control code. It then generates a neutral version of the text, which is assessed by the classifier for its sentiment. Based on the classifier's feedback, the generator refines its output in subsequent iterations, progressively moving towards the neutral sentiment target.

3.4.1. Embedding Layer. The embedding layer serves as the initial stage where the input text is transformed into a high-dimensional vector space. It leverages two types of embeddings: (1) **Word Embeddings:** Each word in the input text is mapped to a unique vector in a predefined embedding space. This mapping can be denoted as $E_{word}: word \mapsto \mathbb{R}^d$, where d is the dimensionality of the embedding space. (2) **Positional Embeddings:** To preserve the sequential nature of text, positional embeddings are added, providing a unique representation for each position in the sequence. It can be represented as $E_{pos}: position \mapsto \mathbb{R}^d$.

The final embedding vector for each token is obtained by summing its word and positional embeddings, i.e., $E_{final} = E_{word} + E_{pos}$.

3.4.2. Encoder with control codes. Our encoder's key innovation is the integration of control codes. Control codes are one-hot encoded vectors representing the desired sentiment state. For instance, considering a three-label sentiment classification task (negative, neutral, positive), a control code for neutral sentiment could be represented as $[0,\ 1,\ 0]$. These codes are concatenated with the input embeddings and processed together, enabling the model to generate text with the intended sentiment. The presence of these codes fundamentally alters the model's focus, directing the generation process towards the desired neutral sentiment.

3.4.3. Decoder. In traditional seq2seq tasks, the decoder generates output sequences one token at a time, relying on previous tokens as context [8]. However, in our model, we deviate from this approach. Our decoder generates the entire sequence in one go, without relying on target sequences. This is achieved through a self-supervised mechanism where the model iteratively refines the generated text based on feedback from the classifier. Techniques like masked self-attention [9] enable the model to focus on relevant parts of the input while generating each word, ensuring semantic content is preserved.

3.5. Self-Supervised Mechanism

The self-supervised mechanism endows our model with the ability to generate more accurate target text and generate text without relying on the target sequence [10]. It is mainly achieved through the classifier we built earlier.

3.5.1. Interaction between Generator and Classifier. Upon generating text, the output is immediately subjected to sentiment classification. The classifier, pre-trained to identify sentiment labels, evaluates the sentiment of the generated text. For instance, if the generator produces a text with a sentiment label of $[0.1,\ 0.8,\ 0.1]$ (representing the probabilities of negative, neutral, and positive sentiments respectively), the classifier assesses how close this output is to the desired neutral sentiment, ideally represented as $[0,\ 1,\ 0]$.

3.5.2. Loss Calculation and Parameter Optimization. The discrepancy between the classifier's output and the neutral sentiment target forms the basis for the loss calculation. Mathematically, this can be expressed using Cross-Entropy Loss:

$$Loss = -\sum_{c=1}^{M} y_{o,c} \log\left(p_{o,c}\right) \tag{4}$$

where M is the number of classes (sentiment labels), y is the binary indicator (0 or 1) of the class label c , and p is the predicted probability of the class label c by the model. The model then backpropagates this loss and updates its parameters through gradient descent to minimize the loss. By iteratively refining its parameters in response to the classification feedback, the generator learns to skew its output towards neutral sentiment.

4. Experiment

4.1. Datasets

The selection and preprocessing of datasets are critical steps in the construction of robust models for sentiment de-stylization. For our study, we strategically chose two datasets, each serving a distinct purpose in training the classifier and the generator components of our model.

4.1.1. Classifier Dataset

To train the classifier, we required a dataset with a substantial representation of neutral sentiments. The Stanford Sentiment Treebank-5 (SST-5) emerged as an ideal choice due to its detailed sentiment categorization [11]. Initially comprising five sentiment classes (Very Negative, Negative, Neutral, Positive, Very Positive), we adapted it into a three-class scheme. Negative and Very Negative, as well as Positive and Very Positive, were merged, aligning with the pragmatic sentiment classification approach typically employed in real-world scenarios, which primarily focuses on the sentiment polarity (Negative, Neutral, or Positive).

We extracted a balanced subset of 2000 samples from SST-5 to ensure a uniform distribution across the sentiment classes. This subset was further divided into 1800 entries for the training set and 200 entries for the testing set. During preprocessing, BERT's tokenizer was utilized to convert the text into token id sequences, ensuring compatibility with the BERT model for accurate sentiment classification.

4.1.2. Generator Dataset. For the generator, we opted for the IMDb movie review dataset, which predominantly contains texts with either positive or negative sentiments. This dataset was particularly suitable for training the generator to focus on the task of sentiment de-stylization. The first 300 entries, arranged in alternating positive and negative reviews, were sampled, with 250 entries allocated for training and 50 for testing.

The dataset underwent meticulous cleaning to remove extraneous characters (like emoticons) and punctuation, ensuring textual clarity. We also performed text padding to standardize the length of text sequences, harmonizing their dimensions in the embedding layer. Additionally, a vocabulary was constructed from the entire IMDb dataset to facilitate index referencing during text generation by the generator.

4.2. Experiment settings

In this study, we initially fine-tuned a classifier built upon the BERT architecture. Specifically, we employed a dedicated classifier training dataset beforehand and conducted three rounds of training to adjust the model parameters for better sentiment recognition. Upon completion of training, we evaluated the model's performance on a test set to ensure its ability to accurately identify different sentiment labels. Subsequently, we applied the well-trained classifier to the self-supervised learning mechanism of the generator. In the configuration of generator, we set the following hyperparameters:

(1) Embedding Size = 256: This determines the dimensionality of the features that the model can capture. A larger embedding layer can aid the model in better understanding and handling the subtle nuances of language.

(2) Number of Transformer Layers = 6: Increasing the number of layers can enhance the model's complexity and its ability to abstract, allowing it to capture deeper linguistic structures. However, this may also increase the risk of overfitting and computational resource requirements.

(3) Number of Attention Heads = 8: The multi-head attention mechanism allows the model to learn information from different subspaces simultaneously, which contributes to the model's flexibility when dealing with complex linguistic phenomena.

(4) Batch Size = 6: This directly impacts the accuracy of gradient estimation and the memory requirements during the training process. Choosing an appropriate batch size can balance training efficiency and model performance.

Our generator, constructed based on the Transformer architecture, required more training rounds to enhance its performance. During the 10 rounds of training, we employed the classifier to supervise the sentiment attributes of the text produced by the generator. In this manner, the generator learned not only how to generate text but also how to adjust the text to fit specific sentiment attributes. Throughout the training process, the model parameters were continuously optimized through backpropagation and gradient descent methods.

4.3. Performance Evaluation

4.3.1. Classifier Evaluation. In our training, the classifier was iteratively trained across three epochs to optimize its ability to recognize sentiment categories. After the first epoch, the model exhibited a loss of 0.665 and an accuracy of 46.1% on the test set. This result indicates substantial initial uncertainty in sentiment classification, demonstrating the model's limited ability to distinguish between sentiment categories. After the second epoch, there was a significant increase in accuracy to 92.4%, with a reduction in loss to 0.388, suggesting that the adjustments and optimizations of the model's parameters were effective. By the third epoch, the accuracy further increased to 96.2%, with a loss decreased to 0.202, indicating that the model could accurately identify different sentiment categories with high classification performance.

Furthermore, we tested the classifier's ability to recognize different sentiment categories. We extracted 100 samples each representing positive, neutral, and negative sentiments from the SST-5 dataset. These samples were fed into the classifier for unlabeled sentiment recognition testing to assess the model's performance in practical applications. The results showed that 98% of the positive samples were accurately classified as positive, with only 2% misclassified as neutral. These misclassified samples were labeled as "positive" in the original dataset, which may indicate that the model has a strong ability to discern between "very positive" and "positive" categories, but there are challenges in identifying borderline positive samples. The recognition results for negative samples were even more accurate, with 99% correctly classified as negative and only 1% misclassified as neutral. The misclassified sample was labeled as "negative" in the original dataset, also hinting that the model might have slight difficulties distinguishing borderline sentiment categories. The accuracy of identifying neutral samples was also high at 98%, with only 2% misjudged as positive, and no samples misjudged as negative. These results demonstrate that the model is highly accurate and robust in distinguishing neutral sentiments from other categories.

In summary, our classifier based on BERT demonstrated high accuracy and reliability when handling sentiment classification tasks, this also demonstrates BERT's strong versatility in the NLP field, as it can adapt to various downstream tasks after fine-tuning. Particularly in distinguishing clear sentiment categories, the model achieved near-perfect recognition. Although there were a few misjudgments for borderline sentiment samples, the overall performance remained stable. These test results provide a solid foundation for our sentiment de-stylization task, ensuring that the subsequent generator can be trained on an accurate sentiment classification basis to simulate neutral texts.

4.3.2. Text De-stylization Evaluation. We did not train separately for the conversion from positive to neutral or from negative to neutral sentiments. Instead, we randomly drew positive and negative samples in each training batch from the dataset. The goal was to enable the model to learn neutral sentiment features more effectively as the Google's Neural Machine Translation System [5] without being influenced by the original sentiment attributes. During the ten rounds of training, the model automatically adjusted its parameters to optimize performance through an internal classifier that discriminates the sentiment tendency of the generated text.

Table 1. Change of the classification loss during different training epochs

Epoch	1	2	3	4	5	6	7	8	9	10
Loss	0.787	0.455	0.410	0.365	0.322	0.288	0.257	0.235	0.215	0.202

During the ten rounds of training, initially, the model's classification loss significantly decreased from 0.787 to 0.455, indicating substantial progress in understanding and generating neutral sentiment text. Subsequently, the model's loss value showed a consistent downward trend, decreasing to 0.410 in the third round, 0.365 in the fourth, and so on until reaching 0.202 in the tenth round, further confirming the effectiveness of the model parameter optimization and the enhancement of its ability to destylize sentiment.

This steady decline in loss suggests that the model's performance in capturing and restoring neutral sentiment is continuously improving. It is noteworthy that the magnitude of the loss reduction slowed down in the subsequent rounds, which may indicate that the model is gradually approaching its learning limit. However, even in the later stages of training, the model was still able to achieve minor performance improvements, demonstrating its robustness in the learning process.

To understand the performance of our model on the task of text sentiment de-stylization, we extracted a subset of data samples from the test set for an in-depth analysis. The selected samples covered texts with positive, negative, and neutral sentiments to assess the de-stylization effect comprehensively. For each sample, we documented the text output by the model and compared it with the original text to observe changes in sentiment attributes. As shown in Figure 2, we have discovered the following phenomena:

(1) **Replacement or Removal of Emotional Vocabulary:** The model tends to replace or remove explicit emotional words when generating neutral text. For instance, a positive word "wonderful" is substituted with a more neutral "little," indicating that the model has learned to identify and reduce the sentiment intensity of the text.

(2) **Role of Attention Mechanism:** The attention mechanism in the Transformer structure helps the model retain the core content and semantic information of the text. Even when some emotional expressions are removed, the model can still capture the dependencies between different positions in the sequence, maintaining the main message of the text.

(3) **Changes in Text Length:** During the de-stylization process, the model sometimes cuts out some descriptive vocabulary, resulting in a reduced text length. This may be related to how the model allocates weight to emotional expressions during learning.

(4) **Impact on Text Readability**: The readability of some samples decreased after removing their emotional tone. This may be due to the model excessively simplifying the text during the de-emotionalization process or because some key coherence vocabulary was mistakenly deleted.

Table 2. Visualization of text samples for text sentiment de-stylization

Label	Input	Output
1 (Positive)	A wonderful little production. The filming technique is very unassuming- very old-time-BBC fashion and gives a comforting, and sometimes discomforting, sense of realism to the entire piece.	A little production. The filming technique is traditional BBC fashion and gives a sense of realism to the piece.
1 (Positive)	I thought this was a wonderful way to spend time on a too hot summer weekend, sitting in the air conditioned theater and watching a light-hearted comedy. The plot is simplistic, but the dialogue is witty and the characters are likable (even the well bread suspected serial killer).	This film offers a way to spend time indoors on a hot summer weekend, sitting in the air conditioned theater and watching a light-hearted comedy. The plot is simple and the dialogue is witty and notable characters.
0 (Negative)	And then we have Jake with his closet which totally ruins all the film! I expected to see a BOOGEYMAN similar movie, and instead i watched a drama with some meaningless thriller spots.	We have Jake and his closet in the film. A BOOGYMAN similar movie is expected to be watched instead of a drama with thriller elements.
1 (Positive)	Petter Mattei's "Love in the Time of Money" is a visually stunning film to watch. Mr. Mattei offers us a vivid portrait about human relations. This is a movie that seems to be telling us what money, power and success do to people in the different situations we encounter.	Petter Mattei's "Time of Money" offers distinct visual elements. The film provides a portrayal of human relations and explores the influence of money, power, and success on people in various situations.
1 (Positive)	Probably my all-time favorite movie, a story of selflessness, sacrifice and dedication to a noble cause, but it's not preachy or boring. It just never gets old, despite my having seen it some 15 or more times in the last 25 years.	This movie presents a story of selflessness, sacrifice, and dedication to a cause, and maintains its appeal over multiple viewings.
0 (Negative)	By 1990, the show was not really funny anymore, and it's continued its decline further to the complete waste of time it is today.	By the 1990, the show is changed. It is continued decreased.

5. Future work

We propose following suggestions to further improve the proposed model. (1) **Incorporation of Pre-trained Text Generation Models.** Our training process did not rely on target sequences to generate text, which may have affected the readability and semantics of the text. We can consider using pre-trained text generation models, such as GPT-2, a language model in multi-tasks learning[12], which would help better capture the semantic information of texts. (2) **Expansion of Vocabulary:** We constructed the vocabulary using the entire training set, but the content of the vocabulary may still be limited. Using more data to build the vocabulary could provide the model with more options for generating neutral emotional texts.

Moving forward, our research endeavors will extend the paradigm of text style transfer, moving past the mere de-stylization of text. Drawing on the methodological insights from "FontGAN" [13] and “Intelligent Typography” [14], which adeptly modulates the visual style of logographic characters, our initiative is to develop a generative framework. This framework will not only sanitize text from its stylistic markers but also endow it with contextually appropriate stylistic nuances. The goal is to facilitate a fluid textual metamorphosis to suit a spectrum of discursive settings, ranging from the rigor of academic discourse to the inventive domains of narrative prose. We plan to harness an array of sophisticated deep learning architectures to ensure meticulous governance over the style modulation process. The objective is to preserve the semantic core of texts while tailoring their stylistic expressions to align with specified communicative intents or brand lexicons [15].

6. Conclusion

In this study, we have developed and trained a BERT-based classifier and a Transformer-based generator for the task of sentiment neutralization in texts. Employing a self-supervised learning approach, the model was trained using texts with negative and positive sentiment attributes to guide the generation of neutral sentiment texts. The experimental results validated the effectiveness of the self-supervised mechanism in the sentiment neutralization task, with the model successfully learning and replicating the characteristics of neutral sentiment. The work presented in this paper not only advances our understanding of self-supervised learning for sentiment neutralization but also lays the groundwork for future applications in more complex text style transfer tasks. Further research could explore the integration of more advanced self-supervised learning techniques to enhance the model's performance and versatility.

Authors contribution

All authors contributed equally to this research, and their names are listed in alphabetical order.

References

[1] Xu W, Long C, Wang R, et al. Drb-gan: A dynamic resblock generative adversarial network for artistic style transfer[C] In Proceedings of the IEEE/CVF international conference on computer vision. 2021: 6383-6392.

[2] Jhamtani H, Gangal V, Hovy E, et al. Shakespearizing modern language using copy-enriched sequence-to-sequence models[J]. arXiv preprint arXiv:1707.01161, 2017.

[3] Olohan M. Intercultural faultlines: research models in translation studies: v. 1: textual and cognitive aspects[M]. Routledge, 2017.

[4] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[C] In Proceedings of naacL-HLT. 2019, 1: 2.

[5] Wu Y, Schuster M, Chen Z, et al. Google's neural machine translation system: Bridging the gap between human and machine translation[J]. arXiv preprint arXiv:1609.08144, 2016.

[6] Li J, Jia R, He H, et al. Delete, Retrieve, Generate: a Simple Approach to Sentiment and Style Transfer[C] In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 2018: 1865-1874.

[7] Andreieva, V. and Shvai, N. Generalization of Cross-Entropy Loss Function for Image Classification [J]. Mohyla Mathematical Journal, 2021, 3, pp.3–10.

[8] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J]. Advances in neural information processing systems, 2014, 27.

[9] Chelba C, Chen M, Bapna A, et al. Faster transformer decoding: N-gram masked self-attention[J]. arXiv preprint arXiv:2001.04589, 2020.

[10] Wen Z, Li Y. The mechanism of prediction head in non-contrastive self-supervised learning[J]. Advances in Neural Information Processing Systems, 2022, 35: 24794-24809.

[11] Ahmet A, Abdullah T. Recent trends and advances in deep learning-based sentiment analysis[J]. Deep learning-based approaches for sentiment analysis, 2020: 33-56.

[12] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners[J]. OpenAI blog, 2019, 1(8): 9.

[13] Liu X, Meng G, Xiang S, et al. Fontgan: A unified generative framework for chinese character stylization and de-stylization[J]. arXiv preprint arXiv:1910.12604, 2019.

[14] Mao W, Su Z, Luo J, et al. A Unified Acceleration Solution Based on Deformable Network for Image Pixel Processing[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2023 (99): 1-1.

[15] Sarker I H. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions[J]. SN Computer Science, 2021, 2(6): 420.