Marvin:
Stylistic Language Editor
Final Report

Vivek Aithal, Priyam Srivastava, Daniel Estevan McAndrew

University of California, Berkeley

May 2021

Abstract

Using contemporary natural language processing (NLP) methods, we have created an AI powered tool for writing called "Marvin". Our project specifically aims to understand how to generate and edit natural language documents to reflect specified styles. Marvin leverages a combination of recent NLP and deep learning innovations such as pre-trained transformer models like BERT (Devlin et al. 2018) and T5 (Raffel et al. 2019) as well as visualization and interpretation techniques like saliency maps (Simonyan, Vedaldi, and Zisserman 2014) and input attribution (Sundararajan, Taly, and Yan 2017). Marvin provides the ability to classify the style of texts, transfer to other styles as well as understand how certain features contribute to style. It strives to adhere to a machine-in-the-loop framework, where writing is performed largely by human users, but is aided by algorithmic suggestions. We evaluated the machine learning models that comprise Marvin on several benchmark metrics and determined that these models are able to perform well on linguistic style tasks. One of Marvin's primary novel contributions is the ability to transfer to different levels of a particular style. It also can perform joint style analysis and transfer for several dimensions of style simultaneously. Such models are lower performing than in the single style case, but still achieve reasonable performance.

Marvin's Synopsis: Style transfer is a really hard task to deal with all the nuance. We think NLP must aid people to write in an appropriate style. To address this problem, we present Marvin. Marvin is a text editor, which allows users to convert text into new styles. It has state-of-the-art advances in the field of ML. This report focuses on our efforts, models, and evaluations.

Links

We have created a site describing the project on Berkeley's School of Information website. We also have been developing the project in this Github repository.

Acknowledgement

We would like to thank Prof. Marti Hearst, our advisor on this project, for her guidance, feedback and kind words of encouragement throughout the course of this semester. We would also like to thank Dongyeop Kang for great insights about the domain of Style Transfer, and technical minutiae of deep learning in NLP. We thank the School of Information for the opportunity and point of views it afforded us, which helped us grow as engineers and practitioners of Artificial Intelligence research. Finally, we would like to thank Douglas Adams for creating the original Marvin the paranoid android and giving us hitchhikers, a guide to the galaxy.

Don't Panic!

1 Introduction

1.1 Motivation

Several theories of language suggest that the meaning of utterances is socially constructed and understood through social contexts (Bakhtin and Holquist 1981). Many factors (like a speaker's gender, race, age, occupation, social status, relationship with their listener, current emotional state etc.) are known to influence how language is structured and used to refer to concepts. One important factor that drives writing style is the domain, since many domains have specific stylistic restraints on how language ought to be structured. Some of these domain-specific restraints can pose significant challenges to new writers. Our tool can aid users in such dilemmas by suggesting edits and additions to meet stylistic constraints. Modern NLP has provided machine language understanding and generation capabilities which are now the frontier of such assistance, going beyond the suggestion of hard-coded "best" practice heuristics as previous tools have done.

At the School of Information we find ourselves at the intersection of cutting edge research and a will to democratize access to such research. We recognize that technology enables us to leapfrog traditional barriers and make powerful tools accessible to all people, enriching our society and culture in the process. The written word has played a vital role in expanding the horizons of human consciousness. And effective use of stylistic language is ubiquitous in this pursuit, from scientific text and Wikipedia to the writings of Shakespeare and Mahatma Gandhi. Unfortunately, language famously suffers from issues of bias, accessibility and quality control. Anyone who has ever made a Wikipedia edit or has written a technical blog post immediately recollects the dozens of examples one has to pore over, to emulate the style, objectivity and tenor of the writing style. Adherence to a particular style can often be used as a proxy measure of value. At best, this leads to a high variance in quality, and at worst, it makes the process of writing entirely prohibitive to newcomers. We believe that a system, such as Marvin, will help users generate style-conformant content and can democratize access to domains for fledgling writers from various backgrounds. This project is a culmination of our journey in the MIMS program, and is heavily heavily informed by our coursework and projects in NLP, machine learning, information retrieval and by our experiences in building user-facing interfaces.

In addition to assisting writers create text appropriate for certain domains, AI-assisted stylistic language editing could help with a number of other applications. For example, activists or whistle-blowers could use such methods to obscure their identity by masking their authorial fingerprint. Screenwriters and novelists could use such models to help develop consistent speaking styles for their characters. Machine translation tasks could leverage these methods to preserve specific styles that may be important for nuanced situations but are often stripped by the translation process.

1.2 What is Style?

All language is uttered in a specific context and those contexts influence a number of its attributes. Style, as the set of non-functional linguistic features of language, comprises some of those attributes. In this way it is distinct from, but sometimes intertwined with the content of language. For example, the two utterances "I'm gonna leave, open the door." and "I wish to depart, could you please open the door?" convey similar content but have different styles. Namely, one might be described as formal, polite, and deferential while the other is informal or perhaps even rude. Style is composed of a number of aspects and dimensions which are at times coupled to one another or to the content.

In addition to more fine-grained styles like politeness, there are other styles like those of specific domains or authors. Such styles establish a unique voice or mood for particular writing. For example, encyclopedias are written in a concrete and declarative language that strives for neutrality and clarity. Similarly, the works of Shakespeare contain a style that is strikingly apparent to modern readers due to its dramatic nature and archaic diction. Contemporary authors also leave similarly distinct stylistic fingerprints. When J.K. Rowling wrote *The Cuckoo's Calling* under the pseudonym "Robert Galbraith", an NLP model was able to detect the author's true identity with high probability ("How Did Computers Uncover J.K. Rowling's Pseudonym?" 2014). We are interested in both of these style scopes in our work, but because of some of the differences between them, we consider them as two different cases.

1.3 Macro-Styles and Micro-Styles

For our project, we denote higher-level domain and authorial styles as "macro-styles". We were initially interested in analyzing such macro-styles. As we further investigated linguistic style to understand what constitutes the macro-styles, we also began to consider lower-level ‘micro-styles’. In some ways macro-styles can be viewed as distributions over a set of micro-styles, and some previous work has modelled them as such. The concept of micro-styles is inspired from Kang et. al (Kang and Hovy 2019) where they define style as a complex combination of different factors, including formality markers, emotions, metaphor, etc. People usually tune these styles in writing differently based on their mood, person they are addressing, content of the message, platform, etc. Multiple micro-styles can jointly describe a given text and distributions over them can define some macro-styles. For example, a given text could be simultaneously funny and sad. Micro-styles also more easily lend themselves to being represented as spectra with varying degrees of intensity. Similar to macro-styles, writers may wish to generate text with certain micro-stylistic properties. For these reasons, we wanted to design our tool to enable users to understand and edit micro-style aspects of their writing. As a proof of concept, we trained models to reason over micro-styles including formality (formal to informal) and emotional valence (happy to sad). Using these models. users can freely tune either single or joint micro-style levels to align with some objective.

1.4 What is Style Transfer?

Recently, there has been a lot of interest in style transfer of text. It is partially motivated from the success of style transfer in images where one can transform images from one style to another. Jing et. al. (Jing et al. 2018) demonstrated the potential of convolutional neural nets (CNNs) in transforming the images from one style to another by separating their content and style. There are numerous mobile applications that can transform images from one style to another. Such applications can, for example, make a photograph look like a van Gogh painting. Learning styles from images is conceptually parallel, albeit simpler problem when compared to text. This is because the extent of features in images are very high and there are consistencies and patterns in the images that can be learnt with enough feature maps. The textual representation is far more discrete, and does not offer the same affordances. However, the idea of separating the content and style is transferable.

A well-implemented style transfer in text would convert the text from one style to another 23 while preserving the content of the text. For example: *I'm crying for your over-kindness.* to *I cry thy overkindness* when transferring from a modern, casual to Shakespearean style. There have been several published papers on textual style transfer in recent years that have taken different approaches. To summarise, there are three main approaches for different use cases (Riley et al. 2020):

1. Supervised Style Transfer:
   It is similar to traditional language translation where one would train a sequential model trained on parallel data of two languages. The constraints here are that one should know both the source and target styles, and have access to the parallel data while training. (Jhamtani et al. 2017) This method is only useful if there is enough parallel data available between two styles, which is often not the case. Due to these constraints, this approach is not scalable.

2. Un-supervised Style Transfer:
   Papers usually refer to this approach as unsupervised, but they are not unsupervised in the true sense. The key difference between 'supervised' and these approaches is that one need not have access to the parallel training data while training the model. But like the supervised approach, the model needs to have access to the source and target style to generate the sentences. (Lample et al. 2019).

3. Arbitrary unknown style to target style transfer:
   This approach views the problem of style transfer such that the model does not have access to the style of the source/input text and need to transfer it to a known target style(Zhao et al. 2018). Access to the parallel data is also not required.

The third approach aligns with the requirement of our tool. We think that as a writer/user of our tool, one will not always know their writing style but would know what style they want the text to be transferred to.

1.4.1 Style Transfer vs Auto-completion

We would like to establish the difference between Marvin and an auto-completion tool. Style Transfer is the task of transforming a given text from one style to another while preserving its content, while auto-completion is the task of predicting the next word in a sentence given the previous word tokens.

In Auto-Completion, the model is trained on huge corpora of text with masked tokens. The objective of the model is to predict the masked token using maximum likelihood estimation. The probability of a token is predicted over the dictionary of all the tokens from text corpora. It can be trained using a Markov Models or a deep learning model like an LSTM (Wang et al. 2020). Auto-completion is forward prediction task conditioned on some previous context. It doesn't have to deal with the consistency of the text content. Its sole objective is to predict the next token correctly. We believe that it is a much simpler problem when compared to the style transfer where the model has to accurately strip the sentence of its style and then replace it with another style while maintaining the meaning of the sentence.

1.5 Related Work

Much of the recent work in style transfer focuses on converting a given sentence of a particular class (as determined by a classifier) into another class without much focus on the semantic content preservation. Lample et al. 2019's work on attribute transfer highlights this fact by changing both the style attribute and the semantic content. As described in the sections above, we are interested in preserving the semantic meaning of sentences while morphing the style. In this section we describe in detail a few relevant related works that informed our understanding of the problem at hand and eventually our own approach. We also discuss the benefits and inadequacies of each approach. We list more related works in the appendix.

- **Seq2Seq or Statistical Methods**

  Reformulating style transfer as a sequence-to-sequence (seq2seq) problem akin to machine translation is a well studied approach. There have been various statistical methods like Xu et al. 2012a or direct seq2seq neural models (Carlson, Riddell, and Rockmore 2018) developed to achieve this goal. Though this works well, this approach is seriously limited by the lack of parallel-aligned datasets for all the styles that we want to transfer between.

- **Plug and Play Language Models (PPLM)**

This is a very exciting paper by Dathathri et al. 2020 from Uber, which uses a discriminator (attribute) model to guide the language model's (LM) generation. This is achieved by a forward and backward pass of the text generated until step t. The gradients from these attribute models tweak the LM's hidden activations to result in the next token that matches the attribute favoring the discriminator. The language model does not require any training and can hence be a generic pre-trained model such as GPT-2, and the discriminator models are far easier to train (with ~ 5 orders of magnitude fewer parameters).

But our experimentation showed that while PPLM does well to complete a given prompt, it fares quite poorly when rewriting a given sentence. We concluded that this approach does not serve our current goals and could potentially be used as a stylistic "auto-complete" feature (without having to fine-tune a separate LM for each style class).

- **Reformulating Unsupervised Style Transfer as Paraphrase Generation**

  In this approach, Krishna, Wieting, and Iyyer 2020a rewrite a sentence in one style to another style, by using an intermediate paraphrase representation. The unique aspect of this paper is that they achieve state-of-the-art results on style transfer without needing parallel datasets in those styles. They first generate a "diverse" paraphrase of the input sentence thus stripping the sentence of its style. Then, the paraphrased sentence is rewritten in the style of choice. Once the paraphrase model is trained (separately on a parallel paraphrase dataset), they use that to create pseudo-gold parallel data for training style models. An important assumption here is that the paraphrase is stripped of its original style and does not leak into the training. The paper addresses this potential issue by training classifiers to predict style on both the original and paraphrased datasets and reporting the accuracy of trained models. This approach ended up largely inspiring the one that we implemented for Marvin's style transfer models.

2 Data

The datasets we used include existing benchmark datasets that we obtained from other researchers as well as datasets that we constructed ourselves from web scraping and text generation.

2.1 Existing Datasets

Cross-Style Language Understanding and Evaluation (xSLUE Kang and Hovy 2019) provides a benchmark corpus for understanding language styles, with a focus on how styles can be dependent on one another and are coupled to content and domain to varying degrees. It contains 15 different styles and 23 classification tasks related to these styles. We used several of these styles and their corresponding datasets during our development. Kang and Hovy 2019 also provides analysis of the associations between different dimensions of style by measuring the correlation of different styles. They do this by using all of the trained classifiers to predict on a new set of texts from tweets and then quantifying the co-occurence. They also sample text from different domains and take a similar approach to understand the stylistic diversity of the selected domains. Ultimately we ended up building our final micro-style models from the formality dataset considered in xSLUE, Grammarly's Yahoo Answers Formality Corpus (GYAFC, Rao and Tetreault 2018). We also used the emotional valence dataset that is part of xSLUE, EmoBank (Buechel and Hahn 2017).

Another existing dataset that we used for performing binary macro-style transfer is the Shakespeare dataset used in Krishna, Wieting, and Iyyer 2020b and assembled by Xu et al. 2012b.

For the existing datasets, we also performed joint classification on the task corresponding to that dataset and also a related task, so we now have pseudo-labels for other related classification problems for some of the datasets. For example, we performed classification of both formality and emotional valence Buechel and Hahn 2017 on the formality dataset, so we now also have emotional valence pseudo-labels for the sentences in that data. We refer to these labels as pseudo-labels because they may be noisy or systematically biased and were generated algorithmically rather than by hand. Despite these qualifications, they could still be useful for understanding joint distributions and coupledness of styles. The table below demonstrates the joint distribution of those styles.

Formality Bucket	Emotional Valence Bucket	Fraction of Data
high	high	0.1762
high	low	0.0616
high	mid	0.0808
low	high	0.1218
low	low	0.1117
low	mid	0.0906
mid	high	0.2137
mid	low	0.0785
mid	mid	0.0650

Table 1: Joint distribution of formality and emotional valence from our dev dataset. It indicates rather surprisingly that these styles are not as tightly coupled as one might expect.

2.2 Newly Assembled Datasets

In addition to some of the standard, available datasets that we used, we also created some of our own datasets via web-scraping and pre-processing.

2.2.1 Abstracts

We created a dataset of scientific abstracts by scraping arXiv. We ended up with 7,161 abstracts written between the years 2010 and 2019 related to the search terms "machine learning", "deep learning", "computer vision", and "natural language processing". We processed these abstracts by splitting them into sentences and removing unusable characters. This gave us a dataset composed of 55,000 sentences which was split into a 50,000 sentence train set and a 5,000 sentence evaluation set.

2.2.2 Wikipedia

Similarly, we scraped Wikipedia for text related to machine learning to build up a dataset with somewhat similar content to the abstracts dataset, but with a different macro-style. We created this article to also contain 55,000 sentences with the same number split between the train and evaluation sets.

Formality Bucket	**Fraction of Data**
mid	0.4101
high	0.2961
low	0.2938

Table 2: Distribution of formality from our dev dataset for the original texts.

Formality Bucket	**Fraction of Data**
high	0.7098
mid	0.2902

Table 3: Distribution of Shakespearean style from our dev dataset for the original texts.

2.3 Pseudo-Parallel Generated Datasets

To train our transfer models, we created a pseudo-parallel dataset of original texts from our data and the results of passing them through a T5 model (Raffel et al. 2019) that we fine-tuned for paraphrasing. We ran the classifiers on the paraphrased text to see if the paraphrasing had the desired effect of stripping stylistic aspects from the text. We also defined 'low', 'mid' and 'high' buckets for the styles so that we could enable the transfer model to transfer to specified levels of a certain style. Based on our investigations, it seemed as though the paraphrasing did mostly have this desired effect.

Other previous work, like Krishna, Wieting, and Iyyer 2020b has found the same phenomenon to hold. The only case where the style was not stripped relatively well was for abstracts. This may be due to the fact that the paraphrases still contained scientific terminology which caused it to be classified as abstract-like. The transfers still seemed to work for this case, though as we treated it as a binary task rather than one with an intra-style axis. For the macro-styles considered, the paraphrase model performed the best on the Shakespearean dataset, which is why we chose that dataset to use not only for binary style transfer but also for intra-style transfer. Below we describe the bucket boundaries and the distribution of the original texts and paraphrased texts as well as statistics describing their differences.

Formality Bucket	**Fraction of Data**
mid	0.1854
mid	0.8146

Table 4: Distribution of Shakespearean style from our dev dataset for the paraphrased texts.

Dataset	**Avg Diff**	**Std Dev Diff**
Formality	0.5178	0.1988
Shakespeare	0.6895	0.4309
Wikipedia	0.5077	0.4984
Abstracts	0.0361	0.1639
Joint Formality/Emo Valence	form: 0.4335, emo: 0.1987	form: 0.2551, emo: 0.2278

Table 5: Absolute difference between original text style scores and their paraphrased outputs.

Bucket definitions:

- Shakespeare : low = [0, 0.1]; mid = [0.1, 0.9]; high = [0.9, 1]
- Formality : low = [0, 0.2]; mid = [0.4, 0.7]; high = [0.9, 1]; skip others
- Emo : low = [0, 0.25]; mid = [0.4, 0.7]; high = [0.9, 1]; skip others

Intervals between the bucket boundaries were skipped for the micro-styles so that the transfer model could better learn to interpret the bucket prompts as coherent entities with defined boundaries. The binary only datasets were not bucketed this way for training the transfer model, since we prompted models to simply transfer to a desired binary style, not to a location along that style axis.

3 Methods

Our style transfer pipeline is inspired by the Krishna, Wieting, and Iyyer 2020b approach of generating a pseudo-parallel dataset using a “diverse” paraphrase model. We then train a model to recover the original sentence from this paraphrase so it learns to transfer to this style from a different style. Classifiers trained on the dataset are used to evaluate the paraphrase model, the transfer models, and also to annotate the intensity of the style. Finally, we obtained 3 different types of style transfer models:

1. Intra-style intensity transfer : Transfer the style of a sentence to a particular level of a given style. For example, make a sentence more formal, or make a sentence more Shakespearean. This is applicable for both micro and macro styles.

2. Joint Style Intensity transfer : Modify the sentence on two or more style axes (formality and emotion, etc.). This is valid in case of micro-styles.

3. Binary Inter-Style transfer : Rewrite a sentence in a preferred style. This is valid in case of macro-styles.

We used the HuggingFace library (Wolf et al. 2020a) for all our training code.

3.1 Style Classification

Marvin contains several classification models which predict the style(s) in which a given text is written. These models enable users to understand how well their current text matches some desired or undesired styles. These models are also essential for other downstream tasks like training intensity transfer models and evaluation.

3.1.1 Joint Sequence Classifier

We trained a joint sentence classification model to classify the sentence on more than one axis (formality and emotion). This joint model comprises of a fully connected layers attached to a pretrained BERT model (Devlin et al. 2018), which acts as an encoder. We then replaced the BERT base with a pretrained DistilBERT model (Sanh et al. 2019) with no effective loss in classification accuracy (but almost 50% reduction in the number of parameters). This single model effectively replaces the need to have a different model for each classification task hence greatly reducing the need for compute resources for both training and inference. It also allows us to create multi-axes pseudo-parallel data for training joint transfer models.

3.2 Text Paraphrasing

We adapted a pre-trained T5 model (Raffel et al. 2020), to generate paraphrases. We trained the model on a the ParaNMT-filtered dataset provided by Krishna, Wieting, and Iyyer 2020b. This is a subset of the ParaNMT dataset with filters applied to promote lexical diversity, syntactic diversity and semantic similarity. This model was then used to generate the pseudo-parallel training data for transfer. We experimented with various sized T5 models and GPT-2, and finally selected the t5-small architecture. This is $\sim 10x$ smaller than the GPT-2 large model used in Krishna, Wieting, and Iyyer 2020b. Based on our experimentation and the recommendation in the appendix of Raffel et al. 2020, we used the "paraphrase: " prefix to train the paraphraser model.

Example - I love to play my guitar and I do not know why
Input - paraphrase: I love to play my guitar and I do not know why
Output - I love playing my guitar and I'm not sure why

Once we had this trained paraphrase model, we used diverse beam search (Vijayakumar et al. 2016) to generate diverse paraphrased outputs.

3.3 Style Transfer

We trained T5 models for the various transfer tasks on pseudo-parallel datasets using the same process described above for the paraphraser. We trained 3 different types of transfer models, using appropriate prefixes. Depending on the task, we used the classifier predictions to rate the chosen style of the original and paraphrase sentences. We then appended this information to the paraphrased sentence in order to achieve the necessary intensity transfers. This method is inspired by the Raffel et al. 2020 in which the authors demonstrate that the T5 architecture is a universal text to text model. They achieve state-of-the-art results in various tasks by simply converting any task into a sequence to sequence task with relevant prefixes and other meta information appended directly to the input text.

1. Intra-style intensity transfer
   Example - what're u gonna do about my issue?
   Goal - Increase formality of the sentence
   Input - transfer: what're u gonna do about my issue? | input: low |

output: high
Output - What do you want to do in relation to my issue?

2. Joint Style Intensity transfer
Example - I'm sad you're going
Goal - Increase formality of the sentence, increase emotion of the sentence
Input - transfer: I'm sad you're going | input formality: low | input emotion :low | output formality: high | output emotion : mid
Output - I am sorry you are going to go.

3. Binary Inter-Style transfer
Example - we found that the architecture of the system has a significant effect on training.
Goal - Convert to a Scientific Abstract style
Input - transfer: we found that the architecture of the system has a significant effect on training. | target: abstract
Output - We found that the system architecture has a very significant effect on the performance of training

3.4 Visualization and Interpretation

We wanted Marvin to allow users to understand how individual tokens contribute to text classification scores. This would enable greater transparency and interpretability as well as provide a way for users to perform a kind of manual style transfer by editing tokens that are pulling a given text's classification away from a desired style. We tried several visualization techniques that considered attention, saliency, and integrated gradients.

Our attempts at using attention-based visualization methods included using a self-attentive model with a single attention mechanism (Lin et al. 2017), using a BERT model with the attention summed over the multiple attention heads. We also considered using transformer attention visualization tools like BertViz (Vig 2019), but this tool is more for understanding the inner mechanics of the model and for sequence-to-sequence outputs.

Our attention summing method worked well in early experimentation (more details of this approach are included in the Appendix) but it only gave an understanding of which tokens were contributing overall to the classifications but did not give an understanding for each class of each task. As we developed joint models, we wanted to generalize the method and allow users to understand how tokens contribute to multiple tasks. Therefore, we developed a method based on saliency maps (Simonyan, Vedaldi, and Zisserman 2014) and integrated gradients (Sundararajan, Taly, and Yan 2017), where we computed the gradients of the model's parameters with respect to each output value in the final output logits and then summed and normalized them over the dimensions to get a representation of the integrated gradient for the embedding representation of each input token.

4 Evaluation

Evaluating stylistic language generation is somewhat difficult. Krishna, Wieting, and Iyyer 2020b provides a review of 23 previous works in this domain and highlights some of the common evaluation approaches as well as their shortcomings. We found their criticisms of previous attempts compelling and followed a similar evaluation approach to what they propose.

There are three main properties that are commonly used for text style transfer tasks: style transfer accuracy, semantic similarity, and fluency (Jin et al. 2020). Style transfer accuracy metrics are meant to quantify how well output texts match the desired style. However, this metric alone is not sufficient. As mentioned, the challenge with style transfer is to not just create text with desired stylistic properties, but also to maintain the content of the original text. Styles vary in the extent to which they are coupled with their content Kang and Hovy 2019 and some of them are inherently tied to content. For example, offensive statements often change their meaning if made unoffensive and it may not make sense to try to simultaneously strip the offensiveness while maintaining the content of many utterances. However, for many styles, including the ones that we consider here, we can attempt to separate semantic content from stylistic aspects and determine how well transferred text retains the original content. For example, taking a transfer that we tested, the statement “i want to see u now” could be transferred to a more formal style while mostly preserving content as “I look forward to seeing you soon!”

A model might achieve a high style transfer accuracy while completely obscuring the original content by deleting relevant information and inserting unrelated features that are characteristic of the desired style. For instance a model that aims to transfer text to a more polite style may just repeat the words “please and thank you”, a meaningless statement that maximizes politeness. Similarly, fluency or “well-formedness” measures the extent to which a given sentence is grammatically correct or interpretable. Models can generate text that preserves content and style but introduces errors or disfluencies, so a fluency metric is needed to understand when this occurs.

To understand these aspects of the quality of our generated texts, we employed both automated machine learning evaluations as well as some early human judgement. We did not directly use these evaluation metrics during model training, but future work may incorporate these metrics into the loss function of the transfer model to generate better results. The human evaluation that we performed is still in an early stage and future work on this problem should involve greater human grading, as human judgement is the best method to understand how our model is performing. Human judgement is also what we aim to ultimately satisfy.

4.1 Style Transfer Evaluation

To evaluate the style transfer accuracy, we used our classifier models. A number of previous studies have taken a similar approach by using classifiers either during training as adversarial discriminators and also to evaluate the transfer accuracy (Yang et al. 2018) (Nogueira dos Santos, Melnyk, and Padhi 2018). Until recently most of these approaches used CNNs like TextCNN (Kim 2014) for this task, but Krishna, Wieting, and Iyyer 2020b used a RoBERTa model (Liu et al. 2019). BERT-like models have been shown to perform better on sentence and document classification tasks (Devlin et al. 2018) (Adhikari et al. 2019) and thus ought to give a better evaluation of style transfer accuracy. We use our classifier models which are custom joint sequence classification models that are based on pretrained DistilBERT Hugging Face models (Sanh et al. 2019) (Wolf et al. 2020b) which have been fine-tuned on our datasets.

We used our classifiers to give a style score to the original sentence, the paraphrased sentence, and the generated sentences for the considered styles. Marvin performs three types of transfers and the ways we use these classification scores differs slightly for each. In each case, generated sentences were meant to have the same style as the original sentence, so we compared the differences in style scores between generated sentences.

In the case of intra-style generation, the model tries to transfer the style of a text from one location on a given style axis to another. For example, we may wish to convert a sentence from low formality to mid-level formality. In this case, we report the difference between the desired style score and the obtained style score (both across all generated sentences and only selecting the best generated sentence). The boundaries of these buckets are described in the Data section. The second case, for joint style transfer over multiple styles is done in an analogous way, but with target styles and buckets for each of the stylistic attribute dimensions.

For the third case of binary style transfer, we want to convert a text that is not from some macro-style domain to one that is. For example, we may wish to convert a text that is not Shakespearean to one that is. An actual model output example of this is a transfer from “Do you want to go hiking tomorrow afternoon?” to “Dost thou desire to hike tomorrow afternoon?”. In this case, we report the transferred style score and also the accuracy for how often the output value is in the highest of a set of buckets.

It is worth noting that while these classifiers are accurate and efficient for performing this analysis, they are of course susceptible to making errors. This is particularly true in cases where there is a possible distribution shift between the original texts and the transferred texts, which can easily introduce false negatives. Similarly, false positives can occur and make the transfer model seem as though it is performing better than it actually is. The quality of the classifier sets an upper bound on the best style transfer accuracy that is obtainable. For example, if a classification model only can attain 90% accuracy, it would rate a perfect transfer output dataset as achieving 90% transfer accuracy, but an imperfect model could surpass this rating due to false positives from the classifier (Jin et al. 2020). We evaluate the classifier models themselves in order to understand how often these misclassification errors occur.

4.2 Content Similarity Evaluation

4.2.1 *n*-gram Based Methods

Many previous style transfer approaches have used BLEU scores (Papineni et al. 2002) for determining content similarity Krishna, Wieting, and Iyyer 2020b. This is the most commonly used metric in these tasks in the literature (Jin et al. 2020), but it has some issues. BLEU was designed to evaluate the quality of machine translations and is largely based off of the n-gram precision for different orders of n. That is, it determines the proportion of n-grams in some candidate translation that are matched in a given reference translation. However, it does not take into account recall, by determining how well n-grams from the reference translations are represented in the candidate, since the concept of recall for comparing multiple references is ill-defined. In order to mitigate this shortcoming and address the problem that short candidate strings that drop information can achieve high BLEU scores, it also contains a brevity penalty to penalize such candidate strings. Despite this penalty, it has been shown that BLEU still does not handle recall well (Banerjee and Lavie 2005). Other text similarity metrics designed for machine translation like METEOR (Banerjee and Lavie 2005) have been shown to correlate better with human evaluations than BLEU (Yamshchikov et al. 2020, Banerjee and Lavie 2005). However, while METEOR addresses some of the shortcomings of BLEU, it is still a method that reasons over aligned mappings between unigrams in the two compared strings. It therefore can still not understand well the issue of synonyms. For example sentences that have very similar meanings to humans but contain few of the actual same words would receive a low METEOR score. Synonym substitutions are some of the most effective ways of maintaining content while shifting the style, so this shortcoming is quite important for style transfer. Newer similarity methods based on word embeddings may be able to more adequately handle the issue of synonyms.

4.2.2 Embedding Based Methods

Based on our consideration of these factors, we decided to adopt an embedding-based similarity metric like that developed in Wieting et al. 2019. We implemented a version of what they have done in that work and developed a similarity score that is a cosine similarity of two sentence embeddings obtained from a pretrained sentence-BERT model (Reimers and Gurevych 2019). Similar to BLEU's brevity score, we scale the cosine similarity of the sentence-level embeddings with a length-difference penalty which is calculated as

$$\mathrm{LP}(s,t) = e^{1-\frac{\max(|s|,|t|)}{\min(|s|,|t|)}}$$

for original sentence s and transfer sentence t where $|t|$ represents the length of t in tokens. Then the overall similarity score is computed as

$$\mathrm{SIM}_\alpha(s,t) = \frac{E(s)\cdot E(t)}{\|E(s)\|\|E(t)\|} LP(s,t)^\alpha$$

where $E(s)$ represents getting the sentence embedding of s and α is a hyperparameter that determines the severity of the length difference penalty punishment. We used $\alpha = 0.4$ as the middle of the range of useful values obtained by Wieting et al. 2019.

We report the average similarity scores between original sentences and the transfers for each of our models over their validation sets. We did not use the semantic similarity metric directly in our model, but only for evaluation. However, using this metric in training may be a good next step for this project in order to obtain better results. Although in general, the content is mostly preserved except for some cases where pronouns are confused in the transfer. This pronoun mix-up seems to be the greatest source of content errors. These errors are hard to detect with automated similarity scores and may need to be evaluated with human judgement. Possibly, a transfer model which first performs part-of-speech tagging like the tag and generate model (Madaan et al. 2020) and penalizes replacing pronouns could address this problem.

4.3 Fluency Evaluation

We have used a query well-formedness score by Salesken.ai (“Query Wellformedness” n.d.) to evaluate the fluency of the data. This model evaluates well-formedness of a sentence which includes a score for non-fragment and grammatically correct sentences. It also penalises for incorrect token capitalization. It is trained on the google well-formed query dataset (“Google Query Wellformedness Dataset” n.d.) which has well-formedness annotations for 25,100 queries from the Paralex corpus (Fader, Zettlemoyer, and Etzioni 2013). However, this dataset might not align with the data distribution of some of our macro-styles like Shakespeare which is mostly based upon archaic English literature. It also tends to penalise informal styles. It works well for macro-styles like Wikipedia, research papers and micro-styles like Formal.

In order to avoid this limitation, we are also tracking the difference in the score for original and transformed sentences at aggregate level. We track the well-formedness score of the original text for each category in a style and check if it aligns with the distribution of the same category after the style transfer.

The fluency results can be found in the Results section.

4.4 Human Evaluations

Human evaluation is the true litmus test of our models. All the previous evaluation methods are automated and metric based. As discussed earlier, we found that there are biases in the dataset on which these evaluation models were trained. For eg: Fluency dataset is biased against the old English literature because of the distribution shift relative to the dataset it was trained on. Also, it is not an ideal metric for informal styles.

Similarity models may not actually match the true intended content of sentences, or at least not as well as humans. These models match how similar sentences are based on word embeddings which should encode content for matching, but still may have some gaps. We found that there were some cases where the text content remained the same, as expected, after the style transfer but the tokens were quite different which led to a low similarity score. Conversely in the case where a single pronoun is flipped by the paraphraser, a transferred sentence may achieve a high similarity score relative to the original, although the meaning has changed notably.

We believe that humans do a better job in understanding the changes in style while maintaining the content. We created a sample dataset of around 100 records of original text, style transfer, and target style (goal of the model). Evaluators were asked to rate the style transfer and content preservation on a 3-point Likert scale.

However, we found that for joint style transfers (like formality-low, emotional valence-mid :: formality-high, emotional valence-high), evaluators might get confused. This may be because the evaluations were done on a single sentence per record which might be too small to comprehend the two dimensions in the text.

The results can be found in the Results section. They are based on the initial round of testing with a handful of human evaluators. We are planning to conduct more sessions of evaluations in the next phase.

4.5 Classifier Evaluation

We also evaluated our classifier models as reported in the Results section. Doing this was much more straightforward then evaluating the transfer models, as we could use standard metrics like the classification F_1 score. These values are reported in the Results section.

5 Results

We developed a number of models throughout our project but we ended up with a handful of final ones that we evaluated and included in our final version of Marvin. These final models include six style transfer models and two joint classification models that each reason over two style-axes.

There were three binary transfer models which attempted to transfer from text that is not written in the desired style, to one that is. These were the scientific abstract, Wikipedia article, and Shakespeare binary models. There was also one macro-style transfer that was done with buckets along an axis from low to high, the Shakespeare model. Similarly, the formality model transfers along an axis for a micro-style. There was also a joint model, which tried to transfer along the two-dimensional style axes of emotional valence and formality.

The transfer models were evaluated based on how well they performed the style transfer, how well they preserved the original content, and how well they resulted in well-formed, reasonably grammatical sentences. The methods for how these models were developed and the mechanics and motivations of these metrics are described in the Methods section.

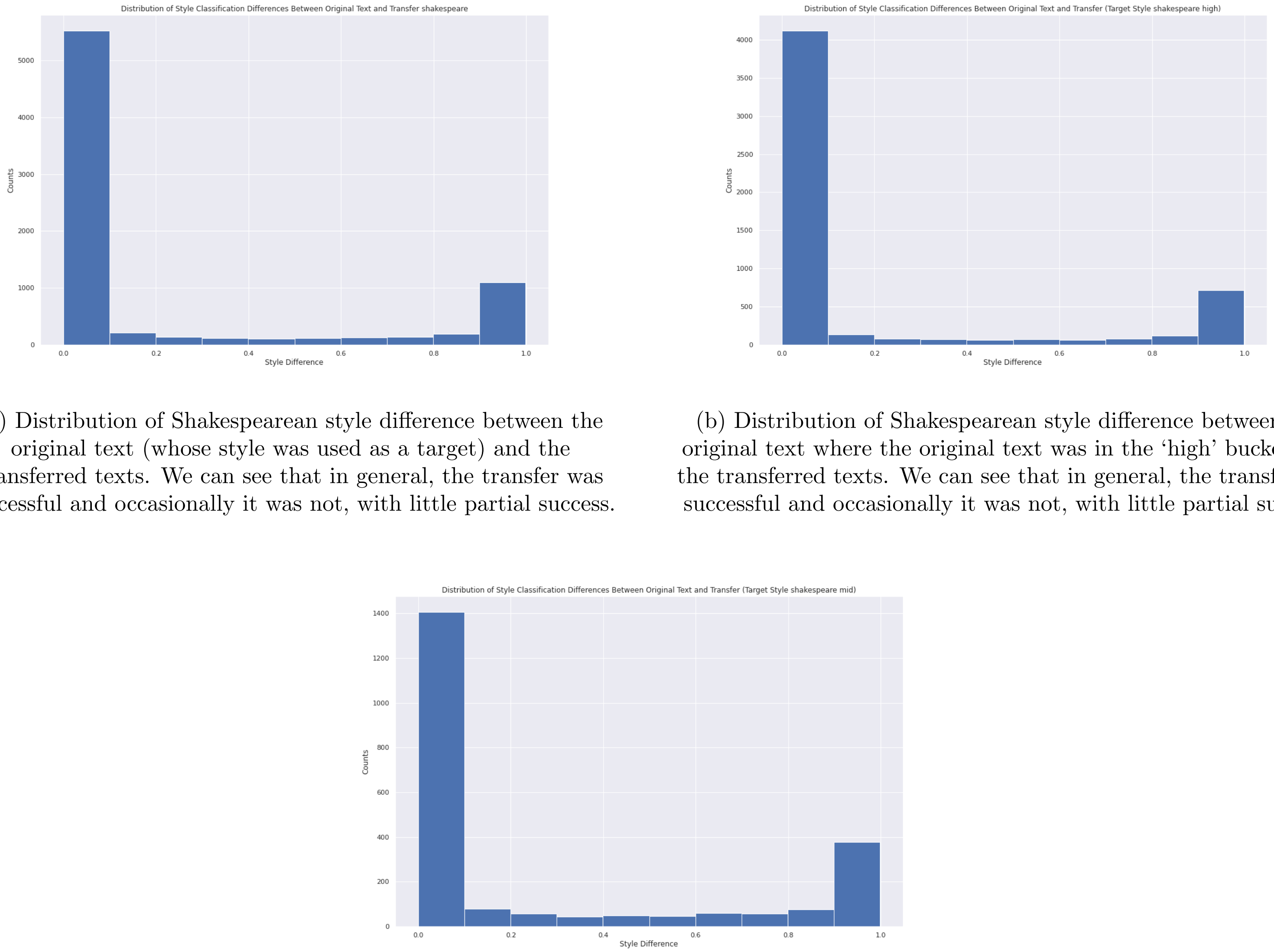
5.1 Style Transfer Accuracy

Model	**Style Diff**	**Bucket Acc**	**Best Bucket Acc**
Abstract	0.003 (0.040)	0.997	0.998
Wikipedia	0.101 (0.301)	0.607	0.815
Shakespeare Binary	0.191 (0.349)	0.847	0.932
Shakespeare	0.216 (0.367)	0.636	0.734
Formality	0.277 (0.250)	0.682	0.860
Micro Joint$_{\text{form}}$	0.441 (0.306)	0.504	0.716
Micro Joint$_{\text{emo}}$	0.215 (0.237)	0.704	0.804

Table 6: Performance of our transfer models on style transfer accuracy. The "Style Diff" column contains the average style score difference between the transferred sentence and the corresponding original sentence written in the target style. These numbers range between 0 and 1 where 0 is better, indicating no difference between the transferred sentence's style and the desired style. In parentheses is the corresponding standard deviation. The "Bucket Accuracy" column contains the fraction of the time that transferred sentences fell in the desired bucket. The desired bucket was determined to be 'high' with a score of at least 0.9 for the binary models. The "Best Bucket Accuracy" is the same metric but considers if at least one of a set of three candidate transfers landed in the desired bucket.

For both Shakespearean style and formality, we see that our models seem to be able to mostly achieve the desired style. In both cases it is more difficult to transfer to the middle of the spectrum than the extreme ends. However, it is an encouraging result that the poorer performance on the middle part of the spectrum seems not too significant.

Based on these results, it seems that our models should allow users to tune the extent to which they want to their text to follow a particular style. The relatively poorer performance of the joint model suggests that this case is notably more challenging and may require more improvements or better datasets to achieve desired performance levels for multi-dimensional cross-style analysis.

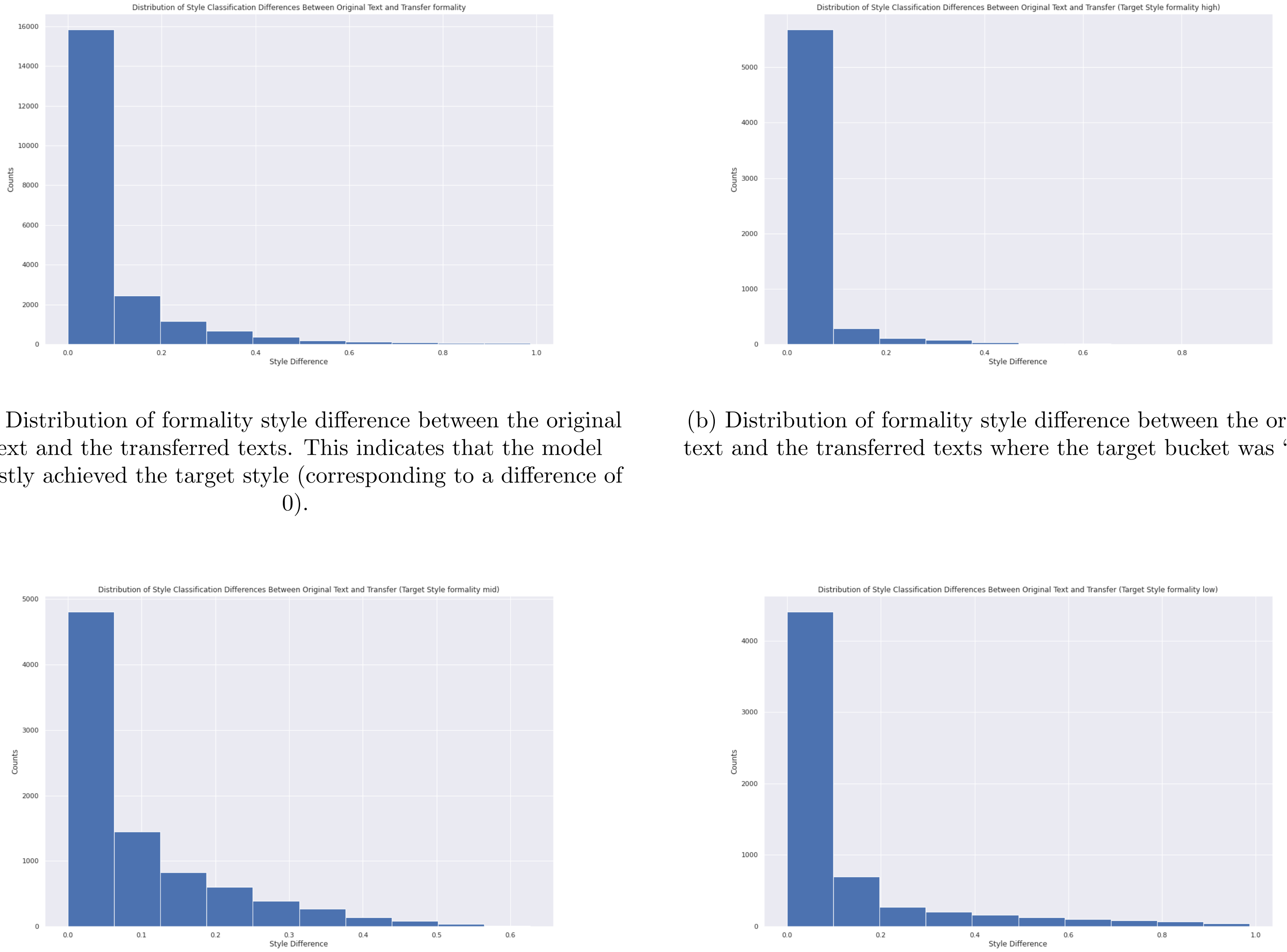


(a) Distribution of Shakespearean style difference between the original text (whose style was used as a target) and the transferred texts. We can see that in general, the transfer was successful and occasionally it was not, with little partial success.

(b) Distribution of Shakespearean style difference between the original text where the original text was in the 'high' bucket and the transferred texts. We can see that in general, the transfer was successful and occasionally it was not, with little partial success.

(c) Distribution of Shakespearean style difference between the original text and the transferred texts where the original text was in the 'mid' bucket.

Figure 1: Shakespeare Style Transfer



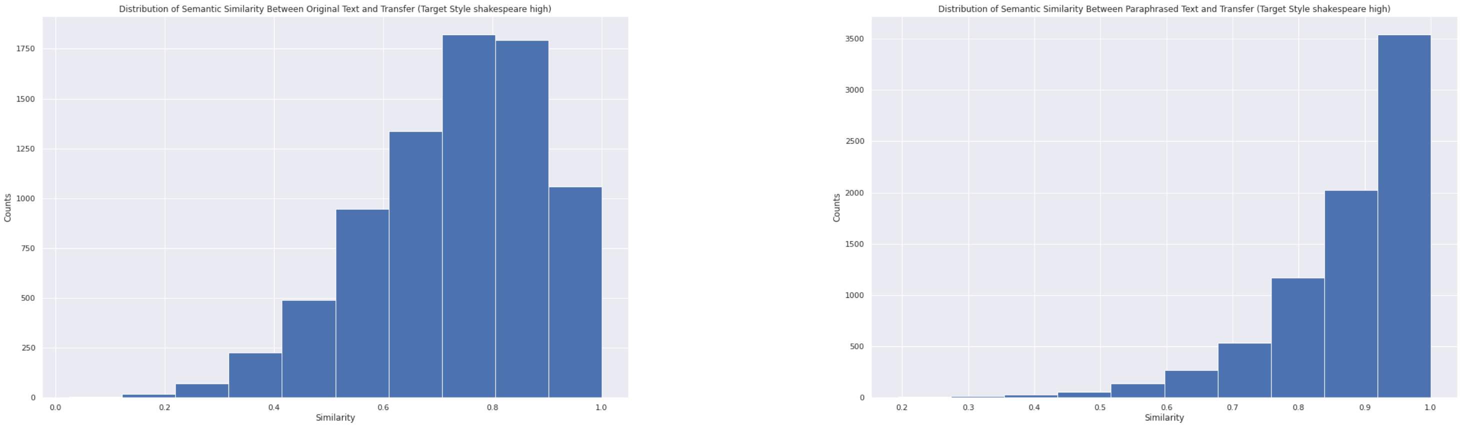
(a) Distribution of formality style difference between the original text and the transferred texts. This indicates that the model mostly achieved the target style (corresponding to a difference of 0).

(b) Distribution of formality style difference between the original text and the transferred texts where the target bucket was ‘high’.

(c) Distribution of formality style difference between the original text and the transferred texts where the target bucket was ‘mid’.

(d) Distribution of formality style difference between the original text and the transferred texts where the target bucket was ‘low’.

Figure 2: Formality Style Transfer



(a) Distribution of transferred text similarity to the corresponding original text.

(b) Distribution of transferred text similarity to the corresponding paraphrase text.

Figure 3: Semantic Similarity of Shakespeare

5.2 Content Similarity

Model	**Orig Sim**	**Para Sim**
Abstract	0.812 (0.114)	0.908 (0.054)
Wikipedia	0.806 (0.114)	0.913 (0.052)
Shakespeare Binary	0.746 (0.156)	0.879 (0.090)
Shakespeare	0.730 (0.162)	0.879 (0.115)
Formality	0.829 (0.145)	0.847 (0.105)
Micro Joint	0.812 (0.146)	0.863 (0.097)

Table 7: Similarity scores for the transfer models. These number range between 0 and 1 and represent how semantically similar the transferred sentences are to reference sentences. “Para Sim” is measure how similar the transferred sentences are to the paraphrased sentence that were used to generate them. “Orig Sim” measure the same thing but for semantic similarity with the original sentence.

We can see from the above semantic similarity plots that while seemingly content was relatively well-maintained throughout the pipeline overall, there was some loss at the paraphrase step, as the transferred texts were closer to the paraphrase that generated them compared to the original. There was also some loss at the transfer step, but seemingly less so.

5.3 Fluency Evaluation Results

Model	Original	Paraphrased	Mean Trans	Max Trans
Abstract	0.59(0.196)	0.75(0.22)	0.82(0.21)	0.75(0.22)
Wikipedia	0.60(0.19)	0.76(0.21)	0.826(0.19)	0.75(0.20)
Shakespeare Binary	0.29(0.23)	0.45(0.26)	0.38(0.22)	0.5(0.26)
Shakespeare	0.48(0.25)	0.27(0.22)	0.53(0.26)	0.41(0.23)
Formality	0.45(0.25)	0.44(0.27)	0.60(0.23)	0.51(0.22)
Micro Joint	0.45(0.26)	0.45(0.25)	0.61(0.24)	0.50(0.22)

Table 8: Well-formedness scores for the transfer models with the original text scores and paraphrased scores for reference. These values range from 0 to 1 where 1 indicates a well-formed sentence and 0 indicates not. The "Mean Trans" column is computed over all of the transfers, while the "Max Trans" takes the best of a set of 3 generated texts for a given input.

5.4 Human Evaluation Results

Model	Style	Content
Abstract	2.97	2.80
Shakespeare	2.65	2.77
Formality	2.44	2.37
Micro Joint	1.94	2.36

Table 9: Results of some human evaluations for similarity and style transfer. The results are averaged over ratings on a scale from 1 to 3 where 1 indicates poor or incorrect performance, 2 indicates partial success but perhaps with some issues and 3 represents successful transfer or preservation of content.

These human evaluations have only been performed for a few evaluators, so further investigation is required to make definitive claims. However, based on these early results, the human evaluations tell a similar story as the automated style and content metrics. They indicate that style is reliably transferred for many of the considered style axes, but less so for the joint model. They also indicate that in general the transfers preserve content but they do occasionally corrupt it.

5.5 Classifiers Results

Model	**F_1 Score**
Formality	0.870
Emotional Valence	0.968
Shakespeare	0.799
Abstract-Wiki	0.999

Table 10: Results of the classifiers. They all achieved about 80% F_1 scores or higher. Abstracts and Wikipedia articles were classified as a single binary classification task, which is why those datasets are reported as a single model. They seemed to be more useful negative classes for each other than other text datasets that we tried, because they are clearly distinguishable but also not trivial such that the model could learn simple heuristics.

5.6 Example Style Transfers

- The synopsis at the end of the abstract was generated by Marvin.
  Input : summary, style transfer's really hard bcoz of all the nuance. we think NLP's gotta help people write in appropriate style. presenting Marvin, to solve this problem. Marvin's a text editor that allows users convert text to a new style. It has the latest advances in ML. this report has our efforts, models n evaluations
  Goal : Convert to Scientific Abstract
  Prediction : Synopsis: Style transfer is a really hard task to deal with all the nuance. We think NLP must aid people to write in an appropriate style. To address this problem, we present Marvin. Marvin is a text editor, which allows users to convert text into new styles. It

has state-of-the-art advances in the field of ML. This report focuses on our efforts, models, and evaluations.

- **Input :** i'm gonna go crazy when i get my OPT card
  Goal : Convert to Formal - high
  Prediction : when I receive my OPT card, I will go crazy.

- **Input :** I purchased a basket of apples for you.
  Goal : Convert to Formal - low
  Prediction : I've bought a apples basket for you

- **Input :** What're you going to do about it?
  Goal : Convert to Shakespeare - high
  Prediction : What will 't be done?

- **Input :** What do you think?
  Goal : Convert to Shakespeare - high
  Prediction : What dost thou say?

- **Input :** I'm sad you've to go
  Goal : Convert to Formal - mid, Emotion - high
  Prediction : I'm sorry you have to go.

- **Input :** machine learning summarized as getting rules from data
  Goal : Convert to Wikipedia
  Prediction : Machine learning is summarized as a way to get rules from data

6 Application

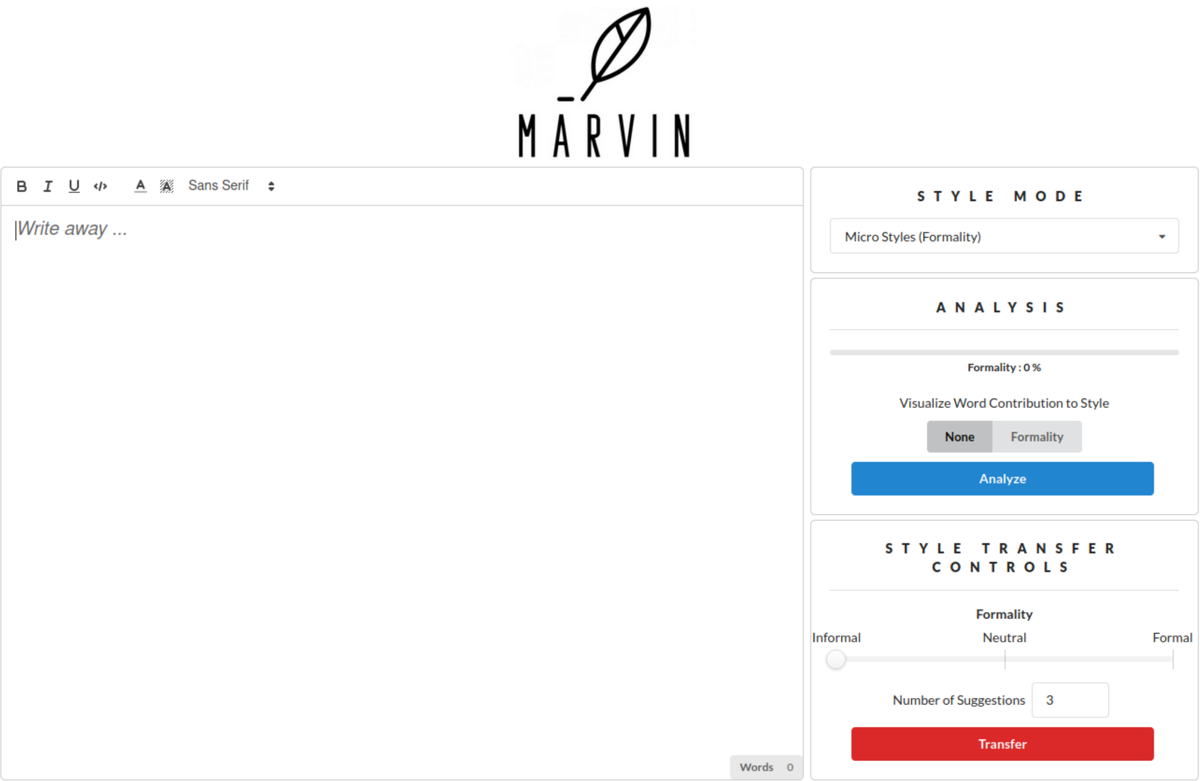
We imagine Marvin belonging to the next generation of text editors. Marvin helps users write collaboratively with state-of-the-art NLP algorithms. Users can not only interpret and analyze their writing and its adherence to various stylistic patterns, but they can also prompt Marvin for suggestions. To operationalize this vision, we developed a web application that can be deployed online.

6.1 User Interface (Front End)

We designed a user interface that allows users to perform the following actions.

- Write using a rich-text editor similar to any modern text application
- Select ”Style Mode” for the application
- View Style statistics of the sentence written
- Visualize the contribution of words towards various styles
- Control the intensity and type of desired style target and get suggestions
- View the style statistics of the suggestions and select a suggestion to replace the current text

We tested our interface informally with several friends and colleagues, and incorporated their feedback.



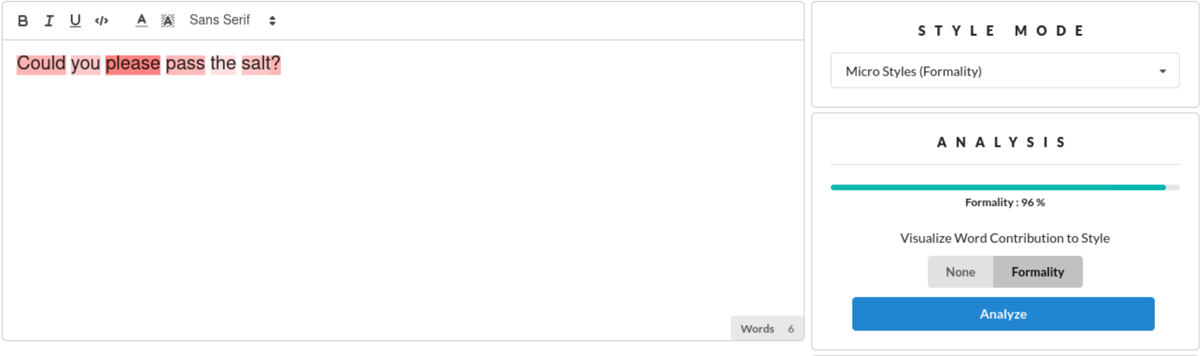
Write away ...

Micro Styles (Formality)

Formality : 0 %

Number of Suggestions 3

Figure 4: Marvin : The interface

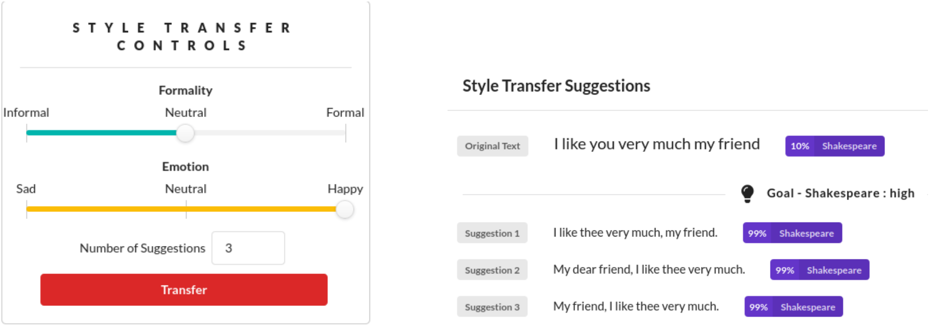


Could you please pass the salt?

Micro Styles (Formality)

Formality : 96 %

Figure 5: Marvin : Analysis and Saliency Visualization



Number of Suggestions 3

Original Text I like you very much my friend 10% Shakespeare

Suggestion 1 I like thee very much, my friend. 99% Shakespeare

Suggestion 2 My dear friend, I like thee very much. 99% Shakespeare

Suggestion 3 My friend, I like thee very much. 99% Shakespeare

(a) Marvin : Style Transfer Controls

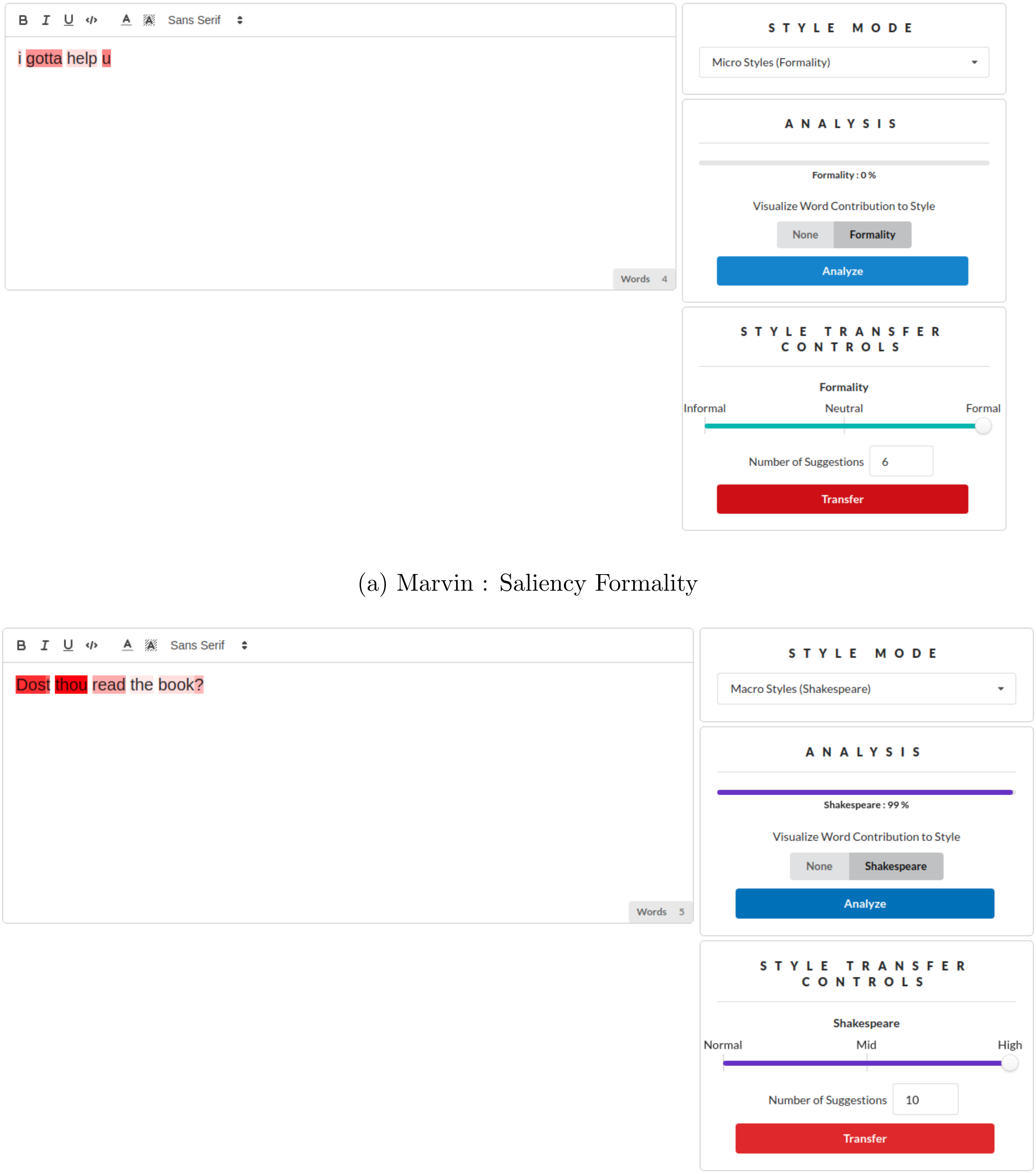
(b) Marvin : Style Transfer Suggestions

Figure 6: Marvin : Style Transfer in action

6.2 APIs (Back End)

We have designed our application with the best practices in back-end engineering with two separate Flask servers. The App server handles all requests related to the application assets and files. All machine learning and model related queries are relayed by the App server to the ML server. This ensures that the application itself is independent of the ML service, which can be swapped out with a better version of the models easily. We have containerized both the servers so that they can be deployed by anyone without any system compatibility issues. Along with APIs for the analysis and transfer, we also have an API for swapping out the models depending on the mode of the application so as to handle the intensive need for computing resources.

We are currently logging all users' interactions with the application into a MySQL database. We hope to use these interactions to generate and further refine our training data, to result in a closed Human-in-the-loop system.



i gotta help u

Words 4

Micro Styles (Formality)

Formality : 0 %

Number of Suggestions 6

(a) Marvin : Saliency Formality

Dost thou read the book?

Words 5

Macro Styles (Shakespeare)

Shakespeare : 99 %

Number of Suggestions 10

(b) Marvin : Saliency Shakespeare

Figure 7: Marvin : Saliency maps texts.

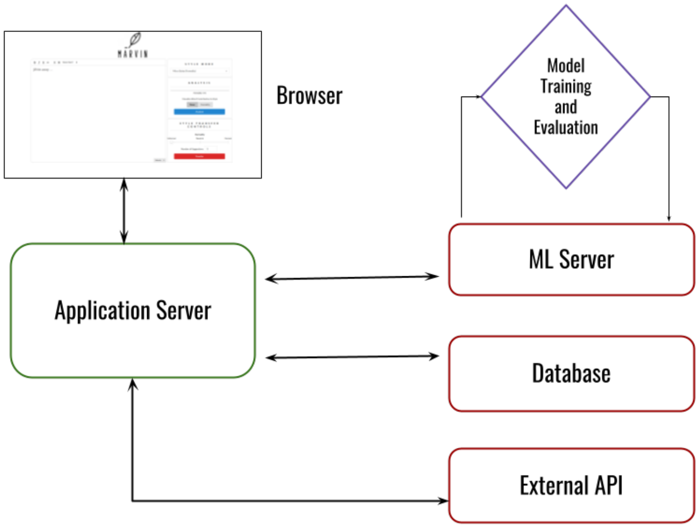


Figure 8: Marvin : Back-end System Architecture

7 Discussion

7.1 Unique Contributions

Despite being inspired by existing work we have cited throughout the document, we believe we have the following unique contributions to the field of Style Transfer.

- We use joint classifiers to predict the probability of a sentence belonging to various style axes at once, reducing the stress of compute resources
- The joint classifiers directly measure the variation of a style along the axis of interest. This not only allows us to perform intra-style intensity transfers, it also enables us to measure the efficacy of the paraphraser model and the transfer model concretely. Most previous work like Krishna, Wieting, and Iyyer 2020b measure the probability of a sentence belonging to a style, among a group of other styles. In this case, the actual variation in a style becomes fuzzy and impossible to measure.
- We introduce the intra-style intensity transfer task, without needing human labelled categories for style intensities. As far as we know, there is no prior work doing this specifically without annotated data.

- We perform joint style transfer to jointly transform micro-styles across two or more dimensions
- Finally, we perform binary transfer for macro-styles in the same model

7.2 Other Advantages of Marvin

- Marvin does not require parallel data. If there is a style classification dataset, we can already perform intra-intensity transfer. If not, we can simply perform a binary transfer nevertheless by just collecting a sufficient dataset of our style.
- We use a DistilBERT backbone for our classifiers, which is almost half the size of similar performing BERT or RoBERTa based models.
- We use a T5 small model, with one-tenth the size of GPT-2 with similar performance.
- We perform robust automated as well as human evaluations to determine the extent of transfer and salience.
- Finally, we have an easy to deploy and use web application to make this project accessible to a wider audience, which aligns with the project's democratizing motivation.

7.3 Future Work

1. One important next step is extending the human evaluations to get a better understanding of model performances. Also if we deploy Marvin as a live web application, we could collect user feedback and use this as a human evaluation metric. We could also use user feedback and behavior information to create labelled datasets to enable better training of models.
2. Kang et. al. (Kang and Hovy 2019) demonstrated 13 different styles that we denote as 'micro-styles'. We have fully-functional models for two of their style axes (Formal-Informal and Happy-Sad) which also work as a joint model with fine tuning. However, as the next phase, we would like to include the remaining styles and offer them as additional dimensions to the user.

3. If we have all the micro-styles included in our tool then the next step would be to provide a pre-adjusted setting of joint micro-style dimensions for macro-styles like Wikipedia, Shakespeare, etc. This will allow users to play with the tuners and create their own desired macro-styles. We believe that this will open avenues for new writing styles in those domains.

4. We previously thought of two approaches that looked promising to achieve our goals. The other approach was to use an end-to-end encoder-decoder network inspired from Zhao et al. 2018 and Riley et al. 2020. It involved training 3 encoder networks and 2 generators. The architecture was such that all the models had to be trained from scratch without using any transfer learning. Thus, this approach was intensive on both time and computation. Due to limited resources, we ended up implementing the current approach. However, we would like to compare its results with our current approach.

5. Human evaluation results showed that it is harder to evaluate joint models (eg: Formality + Emotion) on single sentences. Longer samples seem to be needed to distinguish regions of this joint style space. Similarly, the macro-styles are more apparent in longer sentences/short paragraphs. We would like to incorporate this learning and re-train our models with longer sentences/short paragraphs. We believe that it will improve the performance of our model further.

6. Another possible extension of this work would be to allow users to train models on their own writing samples to understand their own writing style. This could be used to help writers understand what aspects define their writing and they could use this information to make improvements. It could also be used for anonymization of writing.

7.4 Conclusion

We have presented a framework to transform text from one style to another while preserving its content. It doesn't need access to parallel data or style of the given text. We show that a text of any arbitrary unknown style can be paraphrased to a neutral style and then transformed to a target style. We provide models for style transfer to macro, micro, and joint styles.

Most previous works built a generic classifier for all the styles. This classifier is used to measure the style stripping. However, this is an indirect way of measuring the style stripping. We directly measure along the style axis that we are interested in. We show that our approach of measuring the style stripping is better than previous approaches.

We have also built a free and open source tool to demonstrate our work. The user is provided with an interface where they can fine tune the style at both macro and micro levels. They can analyse their text with our 'saliency' feature. They can also use the multi-dimensional style settings for joint styles where one can fine tune styles on multiple axes.

We are very excited about the social impact that our tool can have. We believe that it will democratize access and remove barriers for writers new to a domain. We are optimistic that it will also lead to new style variations in existing domains and might also facilitate writing in styles combinations that have not been explored much. We believe that tools like Marvin are the future of text editing.

8 Appendix

8.1 Additional Examples

Input : He made a new therapeutic technique called logotherapy.
Goal : Convert to Shakespeare, Wikipedia, Abstract
Prediction :
Shakespeare: What dost thou say?
Wikipedia: He created a new therapeutic technique called logotherapy.
Abstract: He developed a novel therapeutic technique called logotherapy

Input : I spelt out some far worse outcomes.
Goal : Convert to Abstract
Prediction : In this paper, I cite some far worse outcomes.

Input : No wonder the sport felt good at that Wimbledon champion's dinner.
Goal : Convert to Formal
Prediction : It's not surprising that the sport felt good at the dinner of Wimbledon champions.

Input : For thousands of years, we have known the perils of getting exactly what you wish for.
Goal : Convert to Shakespeare
Prediction : We have known for thousands of years the perils of getting exactly what thou wish'st.

Input : Can we fix it?
Goal : Convert to Abstract
Prediction : Can we fix the problem?

Input : The problem is right there in the basic definition of AI.
Goal : Convert to Formal
Prediction : The problem is in the basic definition of AI.

Input : In the event of crash, the system shuts off.
Goal : Convert to Abstract
Prediction :he system shuts off when a crash occurs.

Input : No, it's just a silly old skit from SNL.
Goal : Convert to Formal
Prediction : No it's just an old skit of SNL.

Input : yes sure you're are very appealing to me...LOL
Goal : Convert to Formal
Prediction : yes, you're very appealing to me.

Input : Breaking Bad won all the awards at Emmy.
Goal : Convert to Wikipedia
Prediction : Breaking Bad won all the Emmy Awards for its outstanding performance.

Input : He pulled an all-nighter.
Goal : Convert to Formal
Prediction : He pulled a shift for the night.

Input : why do u have to ask such moronic questions?
Goal : Convert to Formal
Prediction : why do we have to ask such questions?

8.2 Other Approaches and Early Work

8.2.1 Related Past Approaches

- **Tag and Generate**

Input	Tagged	Output
send me the text files.	[TAG] send me the text files.	could you send me the text files.
look into this issue.	look [TAG] into this issue.	look forward to hearing from you into this issue.
pass me the salt.	[TAG] pass me the salt.	could you please pass me the salt.
I hate this.	I hate [TAG] this.	I hate to help you with this.
what the hell?	what the hell?	what the hell?

Table 11: Some Tag and Generate examples we experimented with to transfer the "Politeness" style, that highlights the performance and the inadequacies of the system

In this paper, Madaan et al. 2020 perform style transfer in two steps. First, they use a tagger-model to tag all the slots that could be replaced, deleted or inserted in a given sentence to convert the sentence to a desired style, and then they use a second model to fill in these tagged slots. They do not need a parallel dataset to train, since the tagger is first trained on a synthetically created dataset using n-gram tf-idf to identify stylistic phrases. The generator is trained on the earlier created tagged intermediate sentence. This approach, though promising, is limited by its construction. It works well when the only modification needed is one of addition, deletion or replacement and performs poorly if the transfer requires a complete rewrite of the text.

- **Autoencoder**

  This paper by Shen et al. 2017 demonstrates a well-studied approach to style transfer. An encoder takes a sentence and its original style as input and converts it into a paraphrase/style-independent representation. This style-independent representation is then used as input by a generator that generates the sentence in the desired style. This is achieved with an auto-encoder where they introduce a latent content variable z, and latent style variable y for each style. Both the latent variables are being learnt explicitly by the auto-encoder using neural networks for $P(z \mid x, y)$ and $P(x \mid y, z)$. They also add a constraint of consistency in the distribution (cross-alignment) by using the two styles from the same distribution in the input. By using the corpora from same source, they add the constraint that sentences from one style should be similar to the other style from the same population. Here are some examples from our experiments :

1. Positive to Negative sentiment:
   sentence: excellent food and superb service
   transfer: terrible food and poor customer service

2. Negative to Positive sentiment:
   sentence: i am really disappointed
   transfer: i am feeling great

Despite not needing a parallel corpus, this method was suspect as it generates a sentence of a chosen style by learning an aggregate representation of that style. A direct consequence of this is the blurring of lines between style and semantics.

- **TextSETTR**

  This paper by Riley et al. 2020 paper uses a label-free approach of solving the problem of style transfer. The approach taken by Shen et al. 2017 and Krishna, Wieting, and Iyyer 2020a uses non-parallel data but still each corpus has a style associated to it. However, this paper completely eliminates the need for labels. It exploits the fact that there is a connection in style between adjacent sentences of a corpus (unlabeled). While training, they use the adjacent sentences to learn a Style Extractor and an Encoder. As the next step, Encoder conditioned on the Style Extractor tries to recreate the sentence thus learning a decoder. The learned Style Extractor enables us to tune the weights of the style during inference. We believe that is a very significant contribution in the field of Style Transfer in text.

8.2.2 Self Attention and BERT

For our first attempt at classification and attention visualization, we implemented a Self-attentive Sentence Embedding model for text classification

(Lin et al. 2017). This approach introduced a self-attention mechanism that has been very influential in the development of transformers (Vaswani et al. 2017) and the models upon which they are based such as BERT.

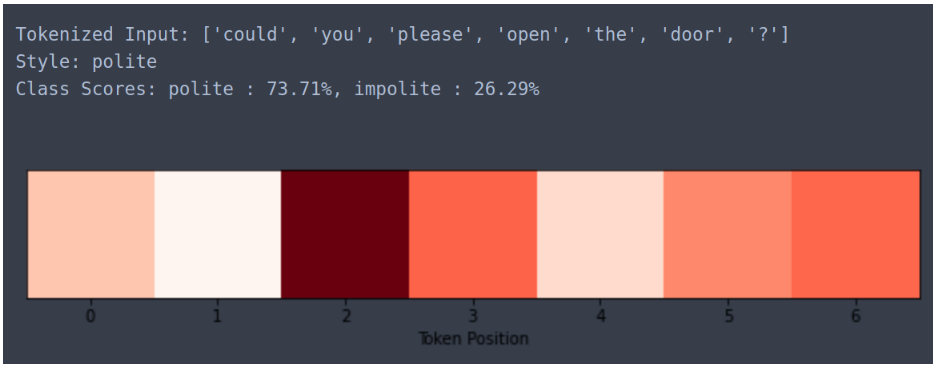
We wanted to use a version of this self-attentive model to classify text as belonging to a certain style and provide users with a visualization of the attention mechanism along with the text's score for each of the style classes. We want users to be able to understand not just which style the model predicted for their text, but also how confident the model is and which tokens contributed most to the style. This model predates transformers and BERT and performs less well than these more recent approaches on many NLP tasks, however, we considered this model because we think that it may be easier for a human to visualize and interpret how the model is arriving at its prediction.

We implemented a version of this model using PyTorch, where a word embedding layer was initialized with weights from GloVe vectors (Pennington, Socher, and Manning 2014). We then trained this model on the Stanford Politeness dataset (Danescu-Niculescu-Mizil et al. 2013) to classify text as either polite or impolite. In our experiments we found that the model was almost achieving a performance on this classification task as that seen in Kang and Hovy 2019, but it was still not quite achieving the accuracy that we wanted. We were able to extract the attention and visualize it though, which was our main goal for using this model. Some of these visualizations made sense, but there were still some shortcomings. On some of our other datasets, such as the Sentiment Treebank (Socher et al. 2013) and the Short-Humor dataset compiled by Kang and Hovy 2019, this approach worked well, but ultimately we thought that a more performant model, while more opaque would be more appropriate.

Because the self-attentive model was not working as well as we wanted initially, we decided to try using BERT models as well. We believed that a BERT based model would perform better on the classification task, but its attention mechanisms might be harder to visualize and interpret. We used a pretrained DistilBERT model (Sanh et al. 2019) from Hugging Face's Transformers library (Wolf et al. 2020b) which had been pretrained on the Sentiment Treebank. We then wrote a pipeline to finetune this model for our datasets.

This DistilBERT model did perform better on the politeness, humor, and sentiment classification tasks than the self-attentive approach and with some of our own adjustments to the model architecture described in the methods, is the classification model that we ended up using. Using the methods described in the visualization section, we were able to create interpretable visualizations of this model's predictions in a similar manner to the earlier attention approach.

8.2.3 Attention Visualizations



```
Tokenized Input: ['could', 'you', 'please', 'open', 'the', 'door', '?']
Style: polite
Class Scores: polite : 73.71%, impolite : 26.29%
```

Figure 9: This is a visualization created while we were developing the heatmap method for visualizing the model's attention and explaining to users what is contributing to the predicted style. Here we were testing with a test sentence not from the dataset which we expected to be polite, ”Could you please open the door?”.

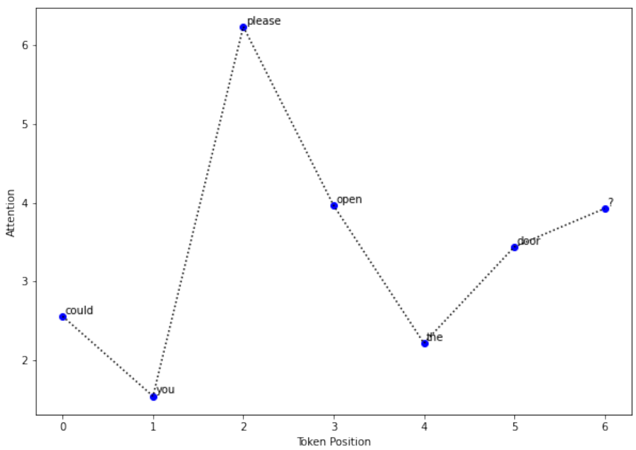


Figure 10: Plot of attention summed over 6 DistilBERT attention heads for predicting the politeness of the sentence ”Could you please open the door?”.

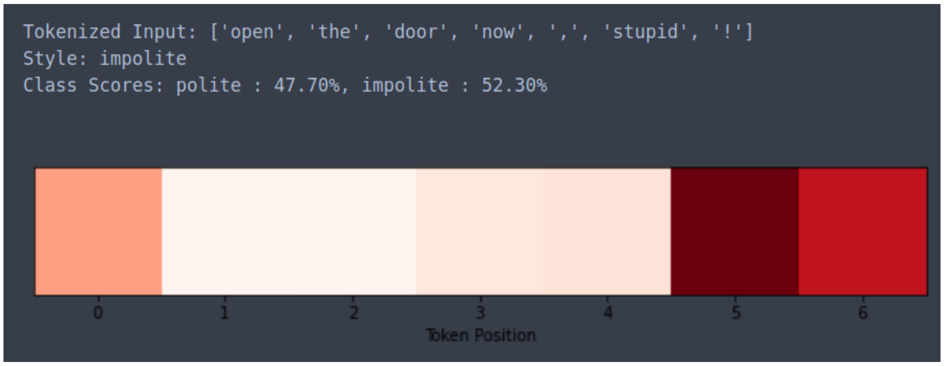

```
Tokenized Input: ['open', 'the', 'door', 'now', ',', 'stupid', '!']
Style: impolite
Class Scores: polite : 47.70%, impolite : 52.30%
```

Figure 11: This is a visualization created while we were developing the heatmap method for visualizing the model’s attention and explaining to users what is contributing to the predicted style. Here we were testing with a test sentence not from the dataset which we expected to be impolite, ”Open the door now, stupid!”

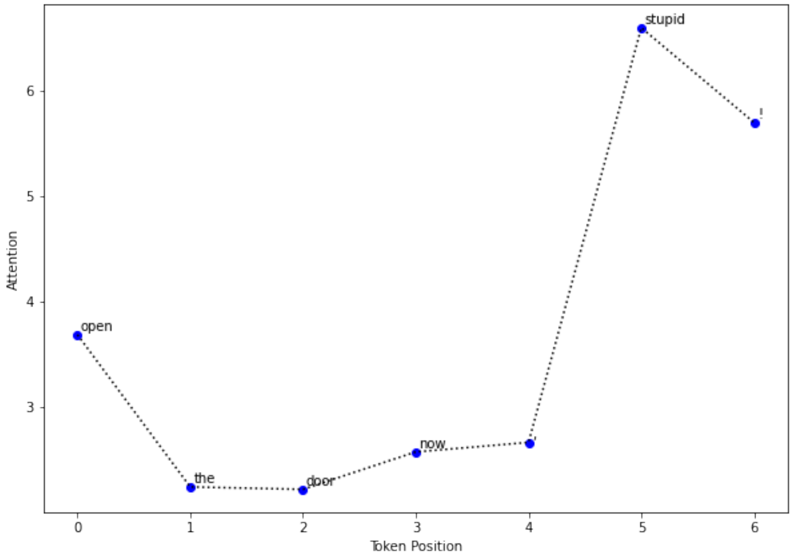


Figure 12: Plot of attention summed over 6 DistilBERT attention heads for predicting the politeness of the sentence ”Open the door now, stupid!”.

References

[] “Google Query Wellformedness Dataset”. In: (). URL: https://github.com/huggingface/datasets/tree/master/datasets/google_wellformed_query.

[] “Query Wellformedness”. In: (). URL: https://huggingface.co/salesken/query_wellformedness_score.

[14] “How Did Computers Uncover J.K. Rowling’s Pseudonym?” In: (2014). URL: https://www.smithsonianmag.com/science-nature/how-did-computers-uncover-jk-rowlings-pseudonym-180949824/.

[Adh+19] Ashutosh Adhikari et al. “DocBERT: BERT for Document Classification”. In: *CoRR* abs/1904.08398 (2019). arXiv: 1904.08398. URL: http://arxiv.org/abs/1904.08398.

[BH17] Sven Buechel and Udo Hahn. “EmoBank: Studying the Impact of Annotation Perspective and Representation Format on Dimensional Emotion Analysis”. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 578–585. URL: https://www.aclweb.org/anthology/E17-2092.

[BH81] M. Bakhtin and M. Holquist. *The dialogic imagination: Four essays*. Austin: University of Texas Press., 1981.

[BL05] Satanjeev Banerjee and Alon Lavie. “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments”. In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72. URL: https://www.aclweb.org/anthology/W05-0909.

[CRR18] Keith Carlson, Allen Riddell, and Daniel Rockmore. “Evaluating prose style transfer with the Bible”. In: *Royal Society Open Science* 5.10 (Oct. 2018), p. 171920. DOI: 10.1098/rsos.171920. URL: https://doi.org/10.1098/rsos.171920.

[Dan+13] Cristian Danescu-Niculescu-Mizil et al. “A computational approach to politeness with application to social factors”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 250–259. URL: `https://www.aclweb.org/anthology/P13-1025`.

[Dat+20] Sumanth Dathathri et al. “Plug and Play Language Models: A Simple Approach to Controlled Text Generation”. In: *International Conference on Learning Representations*. 2020. URL: `https://openreview.net/forum?id=H1edEyBKDS`.

[Dev+18] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805 (2018). arXiv: `1810.04805`. URL: `http://arxiv.org/abs/1810.04805`.

[FZE13] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. “Paraphrase-Driven Learning for Open Question Answering”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 1608–1618. URL: `https://www.aclweb.org/anthology/P13-1158`.

[GEB15] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. “A Neural Algorithm of Artistic Style”. In: *CoRR* abs/1508.06576 (2015). arXiv: `1508.06576`. URL: `http://arxiv.org/abs/1508.06576`.

[Jha+17] Harsh Jhamtani et al. *Shakespearizing Modern Language Using Copy-Enriched Sequence-to-Sequence Models*. 2017. arXiv: `1707.01161 [cs.CL]`.

[Jin+18] Yongcheng Jing et al. *Neural Style Transfer: A Review*. 2018. arXiv: `1705.04058 [cs.CV]`.

[Jin+20] Di Jin et al. “Deep Learning for Text Style Transfer: A Survey”. In: *CoRR* abs/2011.00416 (2020). arXiv: `2011.00416`. URL: `https://arxiv.org/abs/2011.00416`.

[KH19] Dongyeop Kang and Eduard H. Hovy. "xSLUE: A Benchmark and Analysis Platform for Cross-Style Language Understanding and Evaluation". In: *CoRR* abs/1911.03663 (2019). arXiv: 1911.03663. URL: http://arxiv.org/abs/1911.03663.

[Kim14] Yoon Kim. "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. DOI: 10.3115/v1/D14-1181. URL: https://www.aclweb.org/anthology/D14-1181.

[KWI20a] Kalpesh Krishna, John Wieting, and Mohit Iyyer. "Reformulating Unsupervised Style Transfer as Paraphrase Generation". In: *Empirical Methods in Natural Language Processing*. 2020.

[KWI20b] Kalpesh Krishna, John Wieting, and Mohit Iyyer. "Reformulating Unsupervised Style Transfer as Paraphrase Generation". In: *CoRR* abs/2010.05700 (2020). arXiv: 2010.05700. URL: https://arxiv.org/abs/2010.05700.

[Lam+19] Guillaume Lample et al. "Multiple-Attribute Text Rewriting". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=H1g2NhC5KQ.

[Lin+17] Zhouhan Lin et al. "A Structured Self-attentive Sentence Embedding". In: *CoRR* abs/1703.03130 (2017). arXiv: 1703.03130. URL: http://arxiv.org/abs/1703.03130.

[Liu+19] Y. Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *ArXiv* abs/1907.11692 (2019).

[Mad+20] Aman Madaan et al. *Politeness Transfer: A Tag and Generate Approach*. 2020. arXiv: 2004.14257 [cs.CL].

[NMP18] Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. "Fighting Offensive Language on Social Media with Unsupervised Text Style Transfer". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 189–194. DOI: 10.18653/v1/P18-2031. URL: https://www.aclweb.org/anthology/P18-2031.

[Pap+02] Kishore Papineni et al. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.* Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://www.aclweb.org/anthology/P02-1040.

[Pry+20] Reid Pryzant et al. "Automatically Neutralizing Subjective Bias in Text". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01 (Apr. 2020), pp. 480–489. DOI: 10.1609/aaai.v34i01.5385. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5385.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://www.aclweb.org/anthology/D14-1162.

[Raf+19] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *CoRR* abs/1910.10683 (2019). arXiv: 1910.10683. URL: http://arxiv.org/abs/1910.10683.

[Raf+20] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.* 2020. arXiv: 1910.10683 [cs.LG].

[RG19] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, Nov. 2019. URL: http://arxiv.org/abs/1908.10084.

[Ril+20] Parker Riley et al. *TextSETTR: Label-Free Text Style Extraction and Tunable Targeted Restyling.* 2020. arXiv: 2010.03802 [cs.CL].

[RT18] Sudha Rao and Joel Tetreault. "Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 129–140. DOI: 10.18653/v1/N18-1012. URL: https://www.aclweb.org/anthology/N18-1012.

[San+19] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *CoRR* abs/1910.01108 (2019). arXiv: 1910.01108. URL: http://arxiv.org/abs/1910.01108.

[She+17] Tianxiao Shen et al. *Style Transfer from Non-Parallel Text by Cross-Alignment*. 2017. arXiv: 1705.09655 [cs.CL].

[Soc+13] Richard Socher et al. "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. URL: https://www.aclweb.org/anthology/D13-1170.

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *CoRR* abs/1703.01365 (2017). arXiv: 1703.01365. URL: http://arxiv.org/abs/1703.01365.

[SVZ14] K. Simonyan, A. Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *CoRR* abs/1312.6034 (2014).

[Vas+17] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[Vig19] Jesse Vig. "A Multiscale Visualization of Attention in the Transformer Model". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 37–42. DOI: 10.18653/v1/P19-3007. URL: https://www.aclweb.org/anthology/P19-3007.

[Vij+16] Ashwin K. Vijayakumar et al. “Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models”. In: *ArXiv* abs/1610.02424 (2016).

[Wan+20] Sida Wang et al. “Efficient Neural Query Auto Completion”. In: *Proceedings of the 29th ACM International Conference on Information Knowledge Management* (Oct. 2020). DOI: 10.1145/3340531.3412701. URL: http://dx.doi.org/10.1145/3340531.3412701.

[Wie+19] John Wieting et al. “Beyond BLEU:Training Neural Machine Translation with Semantic Similarity”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4344–4355. DOI: 10.18653/v1/P19-1427. URL: https://www.aclweb.org/anthology/P19-1427.

[Wol+20a] Thomas Wolf et al. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 2020. arXiv: 1910.03771 [cs.CL].

[Wol+20b] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[Xu+12a] Wei Xu et al. “Paraphrasing for Style”. In: *COLING*. 2012, pp. 2899–2914.

[Xu+12b] Wei Xu et al. “Paraphrasing for Style”. In: *Proceedings of COLING 2012*. Mumbai, India: The COLING 2012 Organizing Committee, Dec. 2012, pp. 2899–2914. URL: https://www.aclweb.org/anthology/C12-1177.

[Yam+20] Ivan P. Yamshchikov et al. “Style-transfer and Paraphrase: Looking for a Sensible Semantic Similarity Metric”. In: *CoRR* abs/2004.05001 (2020). arXiv: 2004.05001. URL: https://arxiv.org/abs/2004.05001.

[Yan+18] Zichao Yang et al. "Unsupervised Text Style Transfer using Language Models as Discriminators". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/398475c83b47075e8897a083e97eb9f0-Paper.pdf.

[Zha+18] Yanpeng Zhao et al. *Language Style Transfer from Sentences with Arbitrary Unknown Styles*. 2018. arXiv: 1808.04071 [cs.CL].