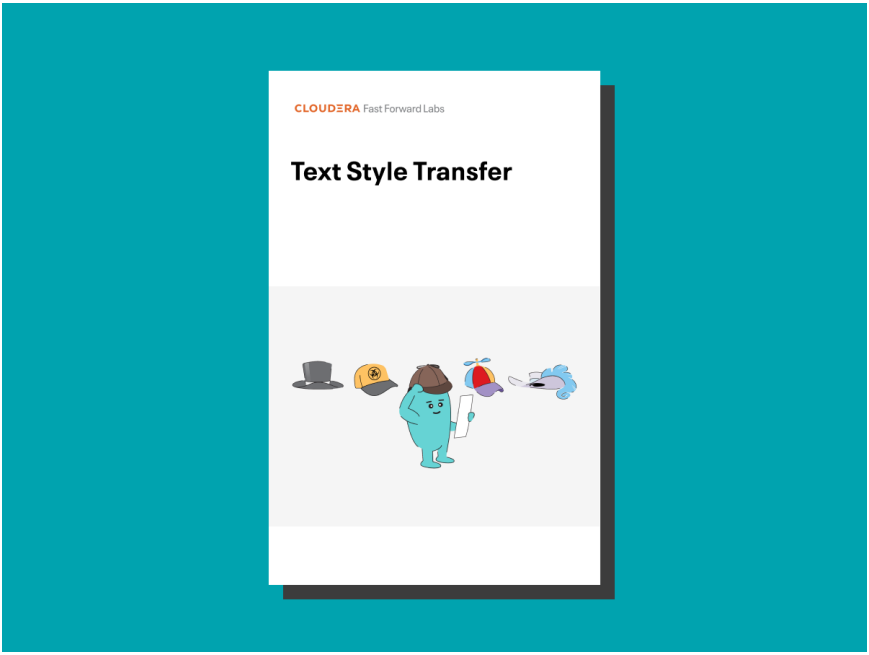
Cloudera Fast Forward

Text Style Transfer

FF24 · *September 2022*



This is an applied research report by Cloudera Fast Forward Labs. We write reports about emerging technologies, and conduct experiments to explore what's possible. Read our full report about Text Style Transfer below, or download the PDF . The prototype accompanying this report – Exploring Intelligent Writing Assistance – demonstrates how the NLP task of text style transfer can be applied to enhance the human writing experience. We hope you enjoy exploring it.

Introduction

Today's world of natural language processing (NLP) is driven by powerful transformer-based models that can automatically caption images, answer open-ended questions, engage in free dialog, and summarize long-form bodies of text – of course, with varying degrees of success. Success here is typically measured by the accuracy (Did the model produce a correct response?) and fluency (Is the output coherent in the native language?) of the generated text. While these two measures of success are of top priority, they neglect a fundamental aspect of language – *style*.

Consider the fictitious scenario where you engage an AI-powered chatbot to assist you with a shopping return for a damaged item. After sharing your intent with the bot, it responds with either of the following generated messages:

1. "Give me a picture of the damage."
2. "Could you please send me a picture of the damage?"

While the first option may contain the correct next action (requesting proof of damage) with sound grammer, something about it feels brash and slightly off-putting in a customer experience setting where politeness is highly valued for customer retention. That's because the expressed tone of politeness plays a critical role in smooth human communication. Of course, this is a non-trivial task for a machine learning model to be aware of as the phenomenon of politeness is rich, multifaceted, and depends on the culture, language, and social structure of both the speaker and addressed person[1].

This quick example highlights the importance of personalization and user-centered design in the successful implementation of new technology. For artificial intelligence systems to generate text that is seamlessly accepted into society, it is necessary to model language with consideration for style, which goes beyond merely just expressing semantics. In NLP, the task of adjusting the style of a sentence by rewriting it into a new style while retaining the original semantic meaning is referred to as *text style transfer (TST)*.

Through this report, we explore text style transfer through an applied use case – neutralizing subjectivity bias in text. We'll start by providing an introduction to TST as a task and its potential use cases. Then, we'll discuss our applied use case, modeling approach, and present a set of custom evaluation metrics for effectively quantifying model performance. Finally, we conclude with a discussion of ethics centered around our prototype: Exploring Intelligent Writing Assistance.

Background

What is text style transfer?

Text style transfer is a natural language generation (NLG) task which aims to automatically control the style attributes of text while preserving the content[2]. To more formally define the task, TST seeks to take the sentence $\mathbf{x_s}$ with source attribute $\mathbf{a_s}$ as input and produce the sentence $\mathbf{x_t}$ with target attribute $\mathbf{a_t}$ that retains the style-independent content of $\mathbf{x_s}$.

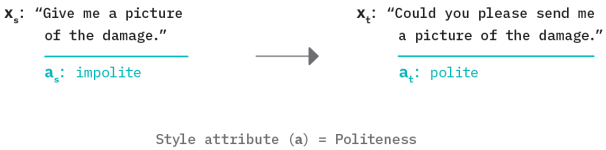


Figure 1: Example of text style transfer that brings impolite language into a polite tone.

Referencing the chatbot example from the introduction where the style attribute is *politeness*, we can see how the style of the input sentence with a source attribute of *impolite* is transferred to have the target attribute of *polite*. Despite the change in tone, the underlying semantic meaning of the two sentences remains largely unchanged.

Of course, politeness isn't the only style attribute which one may seek to control. There is a diverse array of potential style attributes that can be modeled that are largely inspired by *pragmatics* - a branch of linguistics that studies facets of language that are not directly spoken, but rather implicitly hinted or suggested by a speaker and then interpreted by a reader. Within the domain of TST, there are some commonly explored style attributes such as politeness, formality, humor, emotion, toxicity, simplicity, biasedness, authorship,

sentiment, gender, and political slant. The figure below presents illustrative examples for a handful of these common style transfer tasks.

STYLE ATTRIBUTE	SOURCE STYLE	TARGET STYLE
 POLITENESS	POLITE: "**Could you please** send me the data?"	IMPOLITE: "send me the data!!"
 TOXICITY	OFFENSIVE: "I hope they pay out the ***."	NON-OFFENSIVE: "I hope they pay **what they deserve**."
 SIMPLICITY	EXPERT: "Many cause **dyspnea**, pleuritic chest pain, or both."	LAYMAN: "The most common symptoms, regardless of the type of fluid in the pleural space or its cause are **shortness of breath** and chest pain."
 BIASEDNESS	BIASED: "A new downtown is being developed **which will bring back**..."	NEUTRAL: "A new downtown is being developed **which its promoters hope** will bring back..."
 AUTHORSHIP	SHAKESPEAREAN: "My lord, the queen ***would*** speak with you, **and presently**."	CONTEMPORARY: "My lord, the queen ***wants to*** speak with you **right away**."

Figure 2: Illustrative examples of common style attributes.

What is style?

In order to frame a discussion around methods for automatically transferring style between two pieces of text, we must first establish some shared understanding of what *style* actually is and its distinction from *content*. In general, there are two schools of thought for teasing these apart - a *linguistic* definition of style and a *data-driven* definition of style.

The basic idea from a linguistic point-of-view is that a text's style may be defined as *how* the author chose to express their content, from among many possible ways of doing so. We can therefore contrast the *how* of a text (style) from the *what* (content). Linguists look at hand-selected sets of content-independent features (stylistic devices) such as parts-of-speech, syntactic structures, and clause/sentence complexity measures, that in aggregate, can convey a particular style[3]. For example, the style attribute of *formality* is often associated with complex sentence structure, proper punctuation, use of third-

person voice, and exclusion of contractions (e.g. you're, won't) and abbreviations (e.g TV, photos, SKU). While straightforward to interpret, this "rules-based" definition of style actually constrains what can constitute a style (or not a style) to the known set of stylistic devices that exist.

In contrast, the data-driven definition of style assumes a more generalized approach. In this paradigm, given any two corpora, content is the invariance between them, whereas style is the variance[4]. If we think of the linguistic definition of style as a *handcrafted* set of features, then the data-driven definition is a *learned* set of features. This simple definition of style opens the door to a broader range of indicators that may comprise style outside of just those that linguists have terms for. Of course, this comes with a tradeoff in interpretability as we lose the ability to attribute aspects of style to meaningful, explainable linguistic devices (like use of contractions or abbreviations).

This data-driven definition also encompasses more diverse style attribute types including those where style itself is determined not just by linguistic devices, but also by actual words and topic preferences. For instance, if we analyze our chatbot example from earlier, we could intuit that formulating a sentence as a question rather than a statement lends itself to politeness - something that both the linguistic *and* data-driven definitions of style could model. However, we could also correctly intuit that the use of the word "please" is indicative of more polite expression – something that the linguistic approach would exclude (because the word "please" isn't a content-independent feature), but the data-driven approach would capture.

Deep learning architectures commonly used for language modeling today excel at distilling semantic meaning in generalized ways. For that reason, most recent TST work adopts this more encompassing, data-driven definition of style.

Use cases

Text style transfer has many immediate applications in real world use cases today. It also has the potential to support various adjacent NLP tasks like improving data augmentation. Please refer to this survey paper that expands upon the following use cases and more.

Persona-consistent dialog generation

As we've already seen, text style transfer can play a critical role in making human-computer interaction more user-centric. People prefer a distinct and consistent persona (e.g. polite, empathetic, etc.) instead of emotionless or inconsistent persona[5]. Some people appeal more to humor vs. candor vs. drama. TST models could augment NLG pipelines to deliver personalized dialog on an individual user basis.

Intelligent writing assistants

Another industrial application of TST is to enhance the human writing experience. Authors could draft once, but automatically restyle that content to appeal to a variety of audiences - making their ideas more Shakespearean, polite, objective, humorous, or professional.

Text simplification

An inspiring use case for TST is to facilitate better communication between expert and non-expert individuals in certain knowledge domains. For example, automatically simplifying complicated legal, medical, or technical jargon into digestible terminology that a layperson can comprehend, or even lowering language barriers for non-native speakers[6].

Neutralizing subjectivity

Subjective messaging in the form of framing, presupposing truth, and casting doubt is ubiquitous in all forms of writing. For certain texts where objectivity is strongly desired - like news, encyclopedias, textbooks - text style transfer could potentially offer a means to neutralize subjective attitudes[7].

Challenges and considerations

While the idea of modeling the style of text is not new, it has regained attention in the NLP research community with the advent of Transformer models, and consequently a variety of neural methods for automating the task have been recently proposed. This section will explore some of these approaches, along with the challenges and considerations associated with text style transfer in practice.

Availability of usable data

In general, neural methods for TST can be categorized based on whether the working dataset has *parallel* text for a given attribute, or *non-parallel* corpora. Parallel datasets consist of pairs of text (i.e. sentences, paragraphs) where each text in the pair expresses the same meaning, but in a different style. Non-parallel datasets have no paired examples to learn from, but simply exist as mono-style corpora.

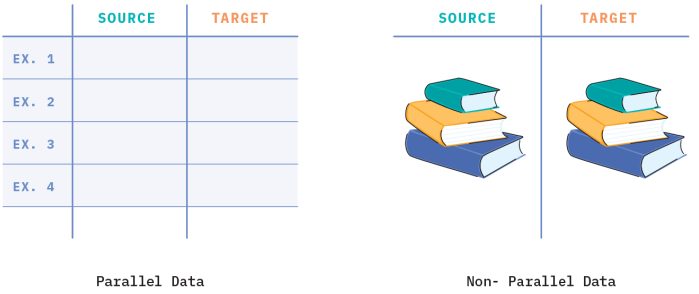


Figure 3: Parallel vs. non-parallel datasets.

For parallel datasets, TST can be formulated similar to a neural machine translation (NMT) problem where instead of translating between languages, we translate between styles. Most approaches adopt some form of a sequence-to-sequence model using an encoder-decoder architecture. While this approach is rather intuitive, the reality is that parallel datasets are rare to find and very difficult to construct. In combination with data-hungry deep learning models that demand copious training examples, obtaining sufficient parallel data for each desired style attribute presents an (often insurmountable) challenge.

Disentangling style from content

Because of the difficulties with parallel data, much of the ongoing research in TST accepts the requirement of only using non-parallel corpora to model style. Without explicit paired examples, the task becomes increasingly difficult. A variety of approaches exist today that fall into three main buckets:

1. **Replacement** - also referred to as “Prototype Editing”, these methods aim to transfer style explicitly by first identifying components (words, phrases,

etc.) of a given sentence that indicate the source style, removing them, and then substituting in new components that represent the target style
2. **Disentanglement** - these methods attempt to implicitly tease apart source attribute style from content in a latent space, and then recombine the content with a new latent representation of style through generative modeling
3. **Pseudo-parallel corpus construction** - tries to reformulate the problem in a supervised manner by creating pseudo-parallel examples from the non-parallel dataset using various tricks such as extracting/matching similar sentences from each corpora as pairs

At the core of all these approaches lies a fundamental question about TST: *Is it actually possible to disentangle style from content? Or is content itself a factor that makes up style?*

It seems the answer somewhat depends on the style attribute being considered and the definition of style adopted. For example, it has been argued that *politeness* is an interpersonal style that can be decoupled from content[8]. In contrast, it feels misguided to say that the style of *sentiment* can be separated from content when altering a sentence's polarity from positive to negative directly changes its semantic meaning.

Overall, this idea of disentangling style from content has been widely discussed in the TST community and remains an open research question[9].

Evaluation

While the descriptions of parallel and non-parallel methods above may be oversimplifications of the actual approaches, it remains apparent how difficult such a task is. To add to the complexity of the problem, TST adopts all of the evaluation challenges faced in general natural language generation tasks, plus some.

To fundamentally evaluate the effectiveness of a NLG output, we must quantify how semantically accurate the generated text was (i.e Did the model say the right thing?), and also how fluent the output is (i.e. Was the thing comprehensible in native language?). The accuracy metric here needs to determine how well the semantic meaning was preserved in the output. For TST specifically, we also need to ensure the target style was achieved. In the end, comprehensive TST evaluation should consider three criteria - transferred style

strength, semantic preservation, and fluency - which often requires human evaluation because automated metrics alone do not adequately characterize these complex properties.

Ethical Concerns

Because text style transfer exists at the crux of generative modeling and personalization, it is imperative that ethical considerations are brought to the forefront of any research agenda. In particular, it's prudent to scrutinize both the beneficial *and harmful* ways in which a technology might be adopted as it may have far-reaching negative consequences.

For example, text style transfer has the potential to help reduce toxicity, hate-speech, and cyberbullying from online social platforms by modeling non-offensive text; a task that currently requires laborious effort via manual content moderation. However, should this technology prove successful, malicious users could just as easily repurpose such methods to model the opposite attribute - generating hateful, offensive text - which counteracts any intended social benefit.

Another example is seen in modeling political slant. A successful endeavor here raises obvious concerns as the ability to automatically transfer attitude and messaging between liberal and conservative tones has the potential to exploit political views of the masses if used for a malevolent social engineering agenda.

These types of task-specific ethical concerns exist in addition to those present with any NLG task – like encoded social bias or generated factual inconsistencies.

Neutralizing Subjectivity Bias

Motivation

Subjective language is all around us – product advertisements, social marketing campaigns, personal opinion blogs, political propaganda, and news media, just to name a few examples. From a young age, we are taught the power of rhetoric as a means to influence others with our ideas and enact change in the world. As a result, this has become society's default tone for broadcasting ideas. And while the ultimate morality of our rhetoric depends on the underlying intent (benevolent vs. malevolent), it is all inherently subjective.

However, there are certain modes of communication today like textbooks, encyclopedias, and [some] news outlets that do strive for objectivity. In these contexts, bias in the form of subjectivity is considered inappropriate, yet it remains prevalent because it is our rooted, societal tone. Subjectivity bias occurs when language that should be neutral and fair is skewed by feeling, opinion, or taste (whether consciously or unconsciously)[10]. The presence of this type of bias concealed within a supposedly objective mode of communication has the potential to wear down our collective trust and incite social animosity as opinions are incorrectly perceived as fact.

Since maintaining a neutral tone of voice is challenging and unnatural for humans, successful automation of this task has the potential to be useful for neutrality-striving authors and editors. Of course, this is no easy feat. Below, we introduce our approach to automatically neutralizing subjectivity bias in text using HuggingFace transformers.

Defining the task

As mentioned earlier, Text Style Transfer (TST) is a natural language generation task which aims to automatically control the style attributes of text while

preserving the content.

In this sense, the task of “neutralizing subjectivity bias” casts subjectivity as the style attribute. Given a subjective sentence, the goal is to generate a modified version of the sentence with the same semantic meaning, but in a neutral tone of voice. In the example below, we see that the source sentence uses the adjective “beautiful” when describing “Newark Academy’s campus”, which is a *subjective intensifier* that implies the author’s feelings about the topic at hand. This sentence can be “neutralized” simply by removing the subjective term as seen in Figure 1 below.

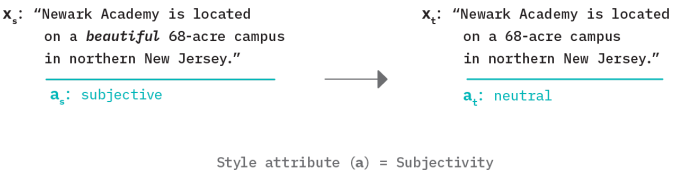


Figure 4: Example of text style transfer that brings inappropriately subjective text into a neutral point of view.

A successful endeavor in this task is predicated on the ability to accurately define and model subjectivity, which is a challenge even for humans because the notion of subjectivity can be... well, subjective. Not all written manifestations of subjectivity bias are this obvious, and consequently, they often cannot be alleviated by a simple rule to remove a modifier word as we’ll see in later examples.

Luckily, there exist open bodies of knowledge like encyclopedias that do adhere to standards of neutral-toned language. For example, Wikipedia strictly enforces a *Neutral Point of View (NPOV)* policy which means representing content fairly, proportionately, and, as far as possible, without editorial bias[11]. To uphold the policy, an active community of editors are incentivized to identify and revise passages that are in violation of NPOV to attain encyclopedic content with a standard tone of neutrality.

Dataset: The Wiki Neutrality Corpus (WNC)

Because Wikipedia enforces this neutrality policy and maintains a complete revision history, the encyclopedia edits associated with an NPOV justification can be parsed out to form a dataset of aligned (subjective vs. neutral) sentence pairs. This realization led to the creation of the Wiki Neutrality Corpus (WNC) – a parallel corpus of 180,000 biased and neutralized sentence pairs along with contextual sentences and metadata – which we will use as the body of knowledge for our TST modeling endeavor. A few examples from WNC are displayed in Figure 2 below.

SOURCE STYLE	TARGET STYLE
A new downtown is being developed which will bring back...	A new downtown is being developed **which its promoters hope** will bring back...
The authors' **exposé** on nutrition studies	The authors' **statements** on nutrition studies
He started writing books **revealing** a vast world conspiracy	He started writing books **alleging** a vast world conspiracy
Go is **the deepest** game in the world.	Go is one **of the deepest** games in the world.
Most of the gameplay is **pilfered from** DDR.	Most of the gameplay is **based on** DDR.
Jewish forces overcome Arab **militants**.	Jewish forces overcome Arab **forces**.
A lead programmer usually spends **his career** mired in obscurity.	Lead programmers often spend **their careers** mired in obscurity.
The lyrics are about **mankind**'s perceived idea of hell.	The lyrics are about **humanity**'s perceived idea of hell.
Marriage is a **holy union** of individuals.	Marriage is a **personal union** of individuals.

Figure 5: Samples from the Wiki Neutrality Corpus that demonstrate sentences before and after neutralization edits are made.

Since the WNC is a parallel dataset, we can formulate our task of “neutralizing subjectivity bias” as a supervised learning problem. In this regard, we indirectly adopt Wikipedia’s NPOV policy as our definition of “neutrality” and aim to learn

a model representation of these policy guidelines directly from the paired examples. But what exactly constitutes Wikipedia's NPOV policy and how are these guidelines realized in practice?

The NPOV policy does not claim to allow *only* neutral facts or opinions. Rather, the goal is to present *all* facts and opinions *neutrally* (without editorial bias), even when those ideas themselves are biased[12]. NPOV advocates the following guidelines to achieve a level of neutrality that is appropriate for an encyclopedia[13]:

- Avoid stating opinions as facts
- Avoid stating facts as opinions
- Avoid stating seriously contested assertions as facts
- Prefer non-judgemental language
- Indicate the relative prominence of opposing views

Upon analyzing actual examples of bias-driven NPOV edits in Wikipedia that result from this policy, the authors of Linguistic Models for Analyzing and Detecting Biased Language and Automatically Neutralizing Subjective Bias in Text observed and categorized several underlying types of bias that appear throughout the WNC: *framing bias*, *epistemological bias*, and *demographic bias.*

Framing Bias

Framing bias is the most explicit form of subjectivity bias and is realized when subjective words or phrases are linked to a particular point of view. As we saw in the example above, the adjective "beautiful" was used to describe the "68-acre campus". These types of subjective intensifiers add directional force to a proposition's meaning, and therefore reveal the author's stance on a particular subject[14].

Epistemological Bias

Epistemological bias results when using linguistic features that subtly presuppose the truth (or falsity) of a proposition and in doing so, modifies its believability. In this way, the author surreptitiously conveys a particular attitude or viewpoint onto the reader in an implicit manner. This type of subjectivity bias is much harder to discern and is often delivered via factive verbs, entailments, assertive verbs, and hedges.

	DEFINITION	SOURCE	TARGET
FACTIVE VERBS	Presuppose the truth of a complement clause.	He **revealed** a multi-national scandal.	He **alleged** a multi-national scandal.
ENTAILMENTS	Are directional relations that hold whenever the truth of one word or phrase follows from another.	He **murdered** the person of interest.	He **killed** the person of interest.
ASSERTIVE VERBS	Are those whose complement clauses assert a proposition.	The theory is a controversial issue, even among conspiracists, many of whom have **pointed out** that it is disproved by...	The theory is a controversial issue, even among conspiracists, many of whom have **said** that it is disproved by...
HEDGES	Are used to reduce one's commitment to the truth of a proposition, thus avoiding any bold predictions.	Eliminating the profit motive **will decrease the** rate of medical innovation.	Eliminating the profit motive **may have a** lower rate of medical innovation.

Figure 6: Common ways in which epistemological bias is surfaced with corresponding examples.

In the first line of Figure 3 above, we see that the term “revealed” is neutralized to “alleged”. This is a clear example of epistemological bias where a factive verb (“revealed”) is used to imply some truth about the subject (“a multinational scandal”), which ultimately may or may not be rooted in fact.

Demographic Bias

Similar to epistemological bias, demographic bias occurs when an author utilizes language that implicitly presupposes truth about people of particular gender, race, religion, or other demographic group. For example, presupposing that all programmers are male through the choice of assigned pronouns[15].

For more detailed discussion on these classes of subjectivity bias, please see this excellent source paper where these definitions and examples are adapted from.

Modeling approach

Now that we have an understanding of the TST task at hand and are familiar with the dataset we'll be using, let's discuss our approach to solving the problem. We will formulate Text Style Transfer as a conditional generation task and fine-tune a pre-trained_BART model on the parallel Wiki Neutrality Corpus in similar fashion to a text summarization use case.

Let's dig into what this means.

Conditional Generation

Recall our goal from earlier: Given a subjective sentence, the goal is to generate a modified version of that sentence with the same semantic meaning, but in a neutral tone of voice.

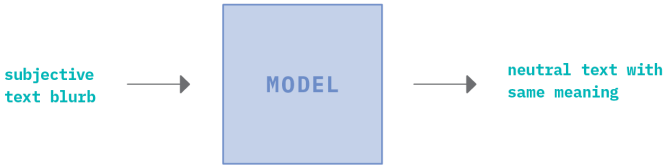


Figure 7: High level TST objective where the goal is to generate output text provided some input text.

By default, we have a generative modeling problem. But how do we go about generating text? It all starts with a language model, which is fundamental to most modern NLP tasks. At its core, a language model is a learned probability distribution over a sequence of words. We can use this probability distribution to estimate the conditional probability of the next word in a sequence, given the prior words as context.

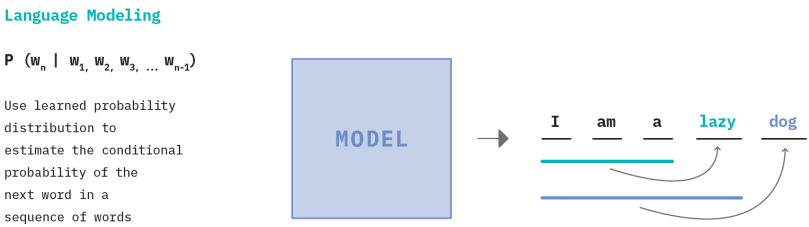


Figure 8: Autoregressive language modeling uses a learned probability distribution to estimate subsequent tokens provided an initial sequence of tokens.

In Figure 8 above, we see that starting with the input sequence “I am a”, the language model is able to iteratively generate subsequent words, one at a time. This describes the fundamental workings of a common NLP method called autoregressive language modeling where the goal is to predict future values from past values (i.e. guess the next token having seen all the previous ones). The notable GPT (and all its descendants) is a popular example of this type of model.

While this is an effective strategy for generating text broadly, it is insufficient for our TST task because for TST we need to generate text that is *conditioned* on our input sentence. Notice that autoregressive models can only generate text a.) based on the statically learned language model and b.) provided an initial sequence of words as a prompt for it to auto-regressively continue on with. In the case of TST, we do not have an initial sequence of a few words as the prompt, rather we have a complete sentence as the prompt that needs to be rewritten from scratch.

What we actually need is a sequence-to-sequence (seq2seq) model to allow for *conditional* text generation.

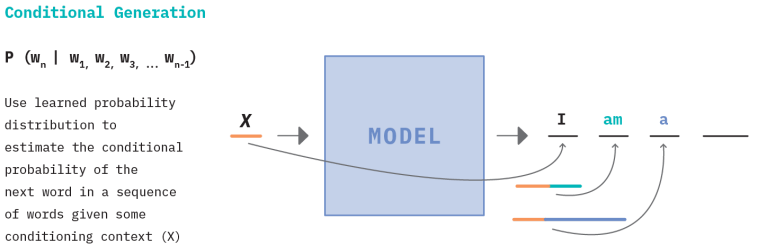


Figure 9: Conditional language modeling uses a learned probability distribution to estimate subsequent tokens conditioned on some input context.

As the name suggests, seq2seq models generate an output sequence conditioned on an input sequence and are the standard class of models for tasks like machine translation, summarization, and abstractive questions answering. In Figure 6 above, we see that the input context **X** is used by the model to generate the first output word ("I"). The generation process then continues in an autoregressive fashion similar to the standard language model, except that for each new term generated, the output is based on the sequence generated thus far *as well as* the input context (**X)**.

This high level discussion helps develop intuition for the general modeling approach (inputs/outputs), but omits many fine details. What is this blackbox language model and how does it "learn" probability distributions over sequences of words? How can it understand the intricate factors that determine subjective vs. neutral language? And how does this model actually condition its outputs based on some input?

We'll answer these questions by taking an indepth look at one particular seq2seq model used in our experimentation called BART and see how it operates as a pre-trained language model.

BART as a Conditional Language Model

Self supervision is a strategy by which models can learn directly from unlabeled data (text in this case), which is crucial for our TST application because we only have a limited number of labeled examples in our parallel WNC corpus. Therefore, self supervised learning (SSL) allows us to first pretrain a model on

enormous bodies of unlabeled text to develop a basic understanding of the English language (i.e. develop a language model). We can then fine-tune this robust representation with the smaller set of parallel training examples from WNC to hone in on the specific patterns attributed to subjective vs. neutral language – a standard process known as transfer learning. For a more detailed review on this topic, see our report FF11: Transfer Learning for Natural Language Processing.

BART is one instance of a model that can be used for self-supervised learning on text data. In particular, BART is a denoising autoencoder that uses a standard Transformer-based architecture for pretraining sequence-to-sequence models, but with a few tricks.

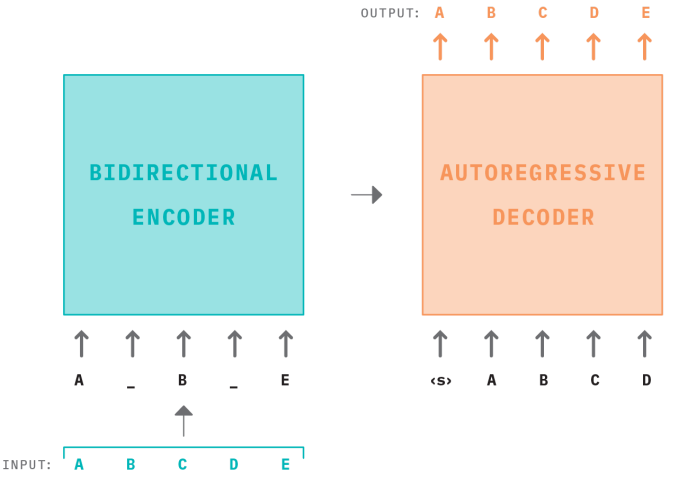


Figure 10: BART is implemented with a bidirectional encoder over corrupted text and a left-to-right autoregressive decoder that attends to the encoded latent representation to generate an output that minimizes the negative log likelihood of the original input document. Image is adapted from the source paper.

BART is pre-trained in a self-supervised fashion on 160GB of news, books, stories, and web text by corrupting input sentences with a noising function and then learning a model to reconstruct the original text (i.e. denoising the

corrupted signal)[16]. The corrupting function works by randomly applying the following transformations (as depicted in Figure 8 below):

- **Text Infilling:** A number of text spans (zero-length, single, or multiple words) are sampled and hidden with a mask token. This teaches the model to predict if, how many, and which tokens are missing from a segment of the sentence.
- **Sentence Permutation:** Input sentences are shuffled in random order in order to teach the model to structure logical statements sequentially.

Figure 11: The two noising transformations that were empirically selected by BART's authors. Image is adapted from the source paper.

In Figure 11 above, we see how noise is first introduced to the input text sequence as tokens are randomly masked. The corrupted document is then processed by a bidirectional encoder (which attends to the full input sequence, forward and backward) to extract out a latent representation of the input. This latent representation gets passed on to the decoder which auto-regressively generates an output sequence conditioned on the latent representation. Reconstruction error is calculated with cross-entropy loss by comparing the original input sequence with the decoder's output (i.e. did the model reconstruct the corrupted input correctly?).

Rather than introducing novel techniques, BART's effectiveness comes from combining the strengths of many advances before it into one empirically driven, cohesive strategy – the architecture of original Transformer, bidirectional encodings from BERT, autoregressive generation from GPT, longer training + larger batch sizes + longer sequences + dynamic masking from RoBERTa, and span masking from SpanBERT.

Establishing a baseline

As with any machine learning problem, it's important to establish a baseline model to serve as a performance benchmark to measure progress against. Upon releasing the WNC dataset, the authors simultaneously released their modeling approach, dataset splits, and results for this TST task, which serve as an excellent benchmark for us to levelset our modeling approach against.

Competitive Benchmark

The paper authors limited their experimentation to a subset of the dataset that only includes NPOV-edits where the Wikipedia editor changed or deleted just a single word in the source text. From our study on the types of bias present in WNC above, we infer that a larger portion of the revisions in this sample consist of framing bias (the most explicit, and therefore easiest type of subjectivity to identify and correct) in comparison to the full WNC corpus. This choice resulted in using just a quarter of the full dataset (~54,000 training pairs) and from that, the authors separated out a random sample of 700 pairs for a development set and 1000 pairs for a test set.

The authors employ two modeling approaches that are both based on an encoder-decoder architecture similar to BART, but with several key differences including the noising function, decoder model type (LSTM RNN vs. attention-based Transformer), and training configuration. For a full description of their modeling setup, see Section 3 of the paper.

The authors quantitatively assess their model performance with two metrics – *BLEU score* and *accuracy*.

1. **BLEU score (bilingual evaluation understudy)** is a common metric used to evaluate the quality of machine translation outputs by looking at the overlap of words and n-grams between a model generated output and a human-generated reference example. BLEU scores range from 0 to 1, where values closer to 1 represent higher degree of similarity.
2. **Accuracy** here is defined as the proportion of all decodings that exactly matched the ground truth references.

Across the two modeling approaches, the author's achieve maximum scores of 93.94 BLEU and 45.80 accuracy on the one-word subset of WNC.

Implementing BART on WNC

The WNC corpus comes cleanly packaged with sentence pairs “pre” and “post” edit for each revision. Prior to modeling the one-word subset with BART, some light exploratory analysis was performed to inform preprocessing decisions.

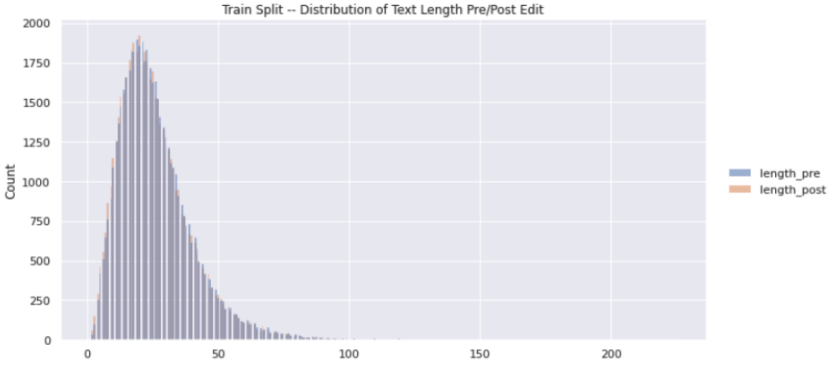


Figure 12: Histogram plot depicting the distribution of text length (both pre and post edit) for all revisions in the training set. Since this plot visualizes both pre and post edit lengths, there are actually ~108k data points represented here (54k each).

The distribution of text length (when naively tokenized by whitespace) is consistent across the provided train/dev/test splits with a median sentence length of 23 tokens. From Figure 9 above, we see that there is a long right tail of sentence pairs by length indicating some potential outliers or data quality issues.

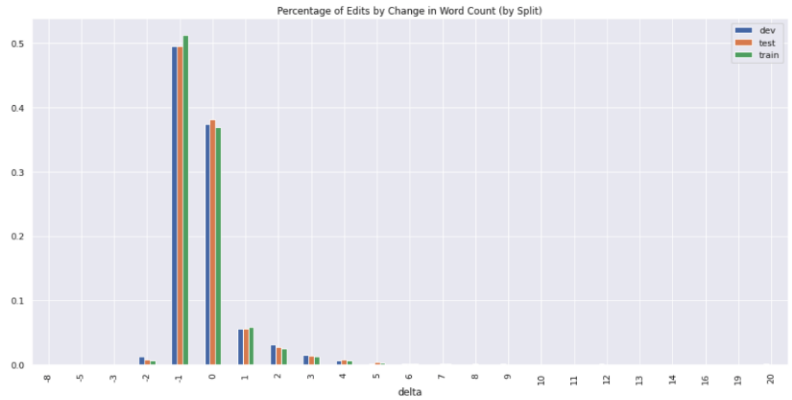


Figure 13: The distribution of revisions by the net change in word count (pre and post edit) expressed as a percentage of total revisions.

When looking at the percentage of sentence pairs grouped by the change in word count before and after the editing (Figure 10), we see that ~50% of revisions are subtractive in nature (i.e. delete one word) and ~38% are net-even in sentence length (i.e. replacing one word). Oddly, we observe some examples where two or more words are removed, which is unexpected given the definition of this subset from WNC.

To account for all of these concerns, the following preprocessing steps are taken:

- Remove records with a pre-edit sentence length above the 99th percentile
- Remove records with a pre-edit sentence length below the 1st percentile
- Remove records with a net subtraction of more than one word
- Remove records with a net addition of more than 4 words (manual inspection shows these are caused by data quality issues like improper punctuation)

For modeling, we make extensive use of the mighty Huggingface transformers library by saving WNC as a HuggingFace dataset, initializing the BartForConditionalGeneration model with facebook/bart-base pretrained weights, and adapting the summarization fine-tuning script for our TST-specific needs. We fine-tune the model for 10 epochs on an NVIDIA Tesla V100 GPU with a batch size of 8, evaluating BLEU and accuracy (with beam search + beam width of 4) every 1000 steps. (Note that when fine-tuning the model with the

parallel examples, the noising function is turned off so an uncorrupted document is passed to both the encoder and decoder.)

The best model from training achieves 93.36 BLEU and 47.39 accuracy. While our results are competitive, they cannot be directly compared with the WNC authors' because of differences in preprocessing. Despite this, they do provide sufficient validation of our approach as a means to automatically neutralize subjectivity bias in text.

Modeling the full dataset

Our efforts thus far in developing a baseline model have affirmed our modeling approach and laid a foundation to improve upon. In the following section, we apply the same data preprocessing steps and model training configuration to the full dataset consisting of ~180,000 subjective-to-neutral sentence pairs that include the one-word edits that we used before, as well as all the sentence pairs with more than one-word edits – a materialy more difficult generative modeling task. We also propose a set of custom automated evaluation metrics aimed to better quantify the subtleties of text style transfer than traditional metrics.

Evaluation Metrics

Challenges with evaluating TST

Evaluating the quality of machine generated text is hard. Human evaluation is regarded as the best indicator of quality, but unfortunately is expensive, slow, and lacks reproducibility, making it a cumbersome approach for validating model performance. For this reason, NLP practitioners often rely on automated evaluation metrics to serve as a cheap and quick proxy for human judgment. Of course, this compromise comes with tradeoffs.

Traditional automated metrics like the BLEU score – the most common metric for evaluating neural machine translation (NMT) – work by counting the lexical n-gram overlap between generated outputs and human-annotated, gold-standard references. As we saw in the previously, BLEU is one of the metrics used by the WNC paper authors to benchmark their model performance against a set of references. Consider the task of comparing the following candidate sentence with the two references while evaluating for semantic equivalence.

Candidate: He is a great singer.

Reference #1: He sings really well.

Reference #2: He is a great writer.

As humans, it's obvious that *Reference #1* means basically the same thing as the *Candidate*, while *Reference #2* changes the entire semantic meaning. However, because BLEU score only measures counts of identical n-grams, *Reference #2* actually scores higher than *Reference #1* by this metric. This highlights one of BLEU's [many] shortcomings in that it fails to robustly match paraphrases, which leads to performance underestimation as semantically-correct phrases are penalized because they differ in lexical form[17].

While it's clear that the BLEU metric itself is flawed, the broader "candidate-to-reference" based NMT evaluation strategy itself also poses issues for evaluating text style transfer. That's because style transfer is a one-to-many task, which

means that there are several suitable references for any one source sentence[18]. Therefore, a high-quality style transfer generation may have a low BLEU score towards a reference as we saw in the previous example. Rather than relying on gold references, *reference-free* evaluation methods have been found to better align with human judgements in the analogous task of paraphrase generation.

In this reference-free paradigm, we ignore ground-truth annotations and compare model output directly with model input. Let's now consider the scenario wherein we feed the sentence "He is a great singer." to our text style transfer model to which it produces an output of "He is a great writer." The first thing we notice is the subjectivity in the sentence has not been neutralized (evidenced by the word "great" in both the input and output sentences) and, worse, the very *meaning* of the sentence has changed – singer and writer are not the same thing!

Unfortunately, BLEU was not designed to detect style, and as we already saw, it's not great at assessing semantics either. We'd end up with a really high evaluation score for a really bad model! For text style transfer, a "one size" score does *not* fit all. We need a comprehensive approach to evaluating TST.

As discussed in our introduction section above, a comprehensive evaluation of quality for text style transfer output should consider three criteria.

1. ***Style strength*** - To what degree does the generated text achieve the target style?
2. ***Content preservation***- To what degree does the generated text retain the semantic meaning of the source text?
3. ***Fluency***- To what degree does the generated text appear as if it were produced naturally by a human?

All three criteria are important in making a determination of quality. If our model transfers text from subjective to neutral tone, but omits or changes an important piece of information (e.g. a proper noun or subject), it fails to preserve the meaning of the original text. On the flip side, if the model reproduces the source text exactly as is, it would have perfect content preservation, but fail completely in style transfer. Finally, the text generation is useless if it contains all the expected tokens, but in an illegible sequence.

Automated evaluation metrics

In the following sections, we'll discuss reference-free, task-specific metrics aimed at tackling the first two of these criteria while also defining our implementation and design choices.

Style Strength

A common automated method for evaluating transferred style strength involves training a classification model to distinguish between style attributes. At evaluation time, the classifier is used to determine if each style transfer output is in fact classified as the intended target style. Calculating the percentage of total text generations that achieve the target style provides a measure of style transfer strength.

While this approach serves as a strong foundation for assessing style transfer, its binary nature means that a quantifiable score only exists in aggregate. The authors of Evaluating Style Transfer for Text improve upon this idea with the realization that rather than count how many outputs achieve a target style, we can capture more nuanced differences between the style distributions of the input and output text using Earth Mover's Distance (EMD)[19]. The EMD metric calculates the minimum "cost" to turn one distribution into the other. In this sense, we can interpret EMD between style class distributions (i.e. classifier output) as the intensity (or magnitude) of the style transfer. Ultimately, this metric called *Style Transfer Intensity (STI)* produces a score that holds meaning on a per-sample, as well as in-aggregate basis.

Implementation

Figure 14 below describes the logical workflow used in our implementation of Style Transfer Intensity.

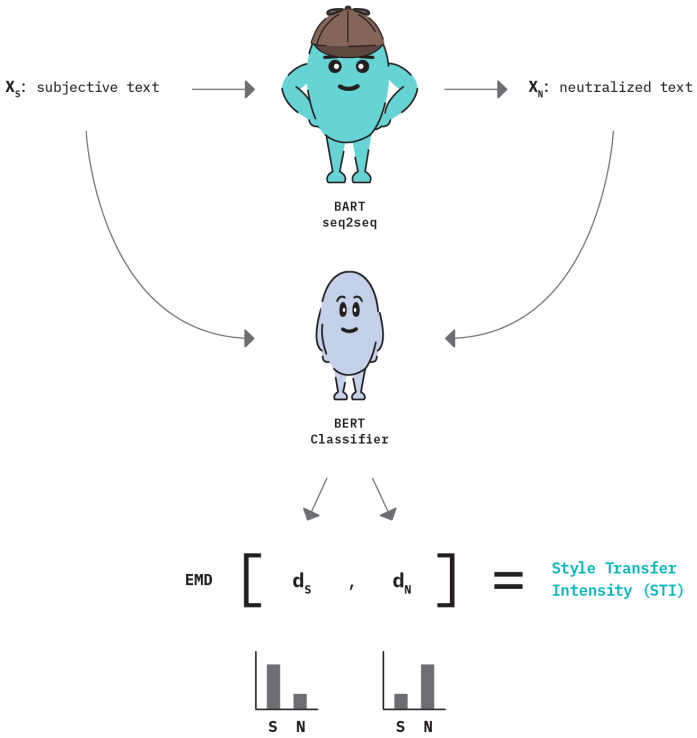


Figure 14: Style Transfer Intensity metric using a BERT classification model.

First (1), a fine-tuned text style transfer model (BART) is used to generate neutralized text *(X_N)* from a subjective input *(X_S)*. This forms the pair of text that we will be calculating the style transfer intensity between.

Then (2) both texts are passed through a fine-tuned, Transformer-based classification model (BERT) to produce a resulting style distribution for each text (d_S, d_N). These style distributions can be visualized at the bottom of Figure 1.

Finally (3), Earth Mover's Distance is calculated on the two distributions to produce a resulting STI score. Note that like the original paper author's, we penalize STI by negating the EMD score if the output text style distribution moves further away from the target style.

Fine-tuning the BERT Classifier

The BERT model from (2) has been fine-tuned on the same style classification task for which the style transfer model was also trained on. In this case, that means reformatting records in WNC from *source_text* | *target_text* pairs into *source_text: subjective; target_text: neutral* labels. In doing so, we maintain the same data splits (train/test/validation), but double the number of records in each split since each sentence pair record from the style transfer dataset becomes two independent examples in the classification dataset.

For training, we initialize HuggingFace's AutoModelforSequenceClassification with bert-base-uncased pre-trained weights and perform a hyperparameter search over: batch size [16, 32], learning rate [3e-05, 3e-06, 3e-07], weight decay [0, 0.01, 0.1] and batch shuffling [True, False] while training for 15 epochs.

We monitor performance using accuracy as we have a perfectly balanced dataset and assign equal cost to false positives and false negatives. The best performing model produces an overall accuracy of 72.50% and has been published to the HuggingFace model registry for experimental use – please reference our training script and classifier evaluation notebook for further details.

Content Preservation

Measuring content preservation between input and output of a style transfer model is often likened to measuring document similarity. As we've mentioned, there are numerous techniques used to quantify similarity between text including traditional lexical-based metrics (e.g. BLEU, METEOR, ROUGE) and newer embedding-based metrics (e.g. WMD, MoverScore, SBERT). However, content preservation in the context of reference-free text style transfer evaluation is uniquely challenging. That's because these similarity metrics fail to account for the aim of style transfer modeling, which is to alter style by necessarily changing words. Therefore, intended differences (changes in style) between source and target text are often incorrectly penalized[20].

To evaluate content preservation more precisely, attempts have been made to first distinguish between semantic and stylistic components of text, and then meaningfully quantify the similarity of just the semantic component alone. While there is open debate about whether it's possible to actually decouple style from content in free text, intuition leads us to believe that our style attribute of "subjectivity" is expressed, at least in part, through select words. For example, our EDA findings have shown that the presence of certain modifiers (adjectives and adverbs) are strong indicators of subjective content.

Previous efforts have approached this style disentanglement process by isolating just the content-related words in each sentence (i.e. masking out any style-related words). They do this by training a style classifier and inspecting the model for its most important features (i.e. words). These strong features form a *style lexicon*. At evaluation time, any style-related words from the lexicon that exist in the input or output texts are masked out – thus leaving behind only content-related words. These "style-free" sentences can then be compared with one of the many similarity measures to produce a content preservation score.

We draw inspiration from the aforementioned tactic of "style masking" as a means to separate style from content, but implement it in a different manner.

Implementation

Rather than construct a global style lexicon based on model-level feature importances, we dynamically calculate local, sentence-level feature importances at evaluation time. We prefer this method because the success of the Transformer architecture has shown that contextual language representations are stronger than static ones. This approach allows us to selectively mask style-related tokens depending on their function within a particular sentence (i.e. some words take on different meaning depending on how they are used in context) instead of relying on a contextually-unaware lexicon lookup.

We accomplish this by applying a popular model interpretability technique called Integrated Gradients to our fine-tuned BERT subjectivity classifier which explains a model's prediction in terms of its features. This method produces *word attributions*, which are essentially importance scores for each token in a sentence that indicate how much of the prediction outcome is attributed to that token.

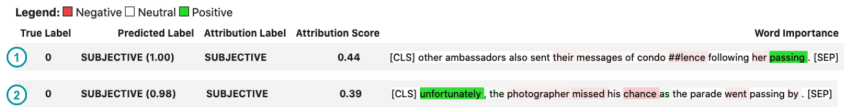
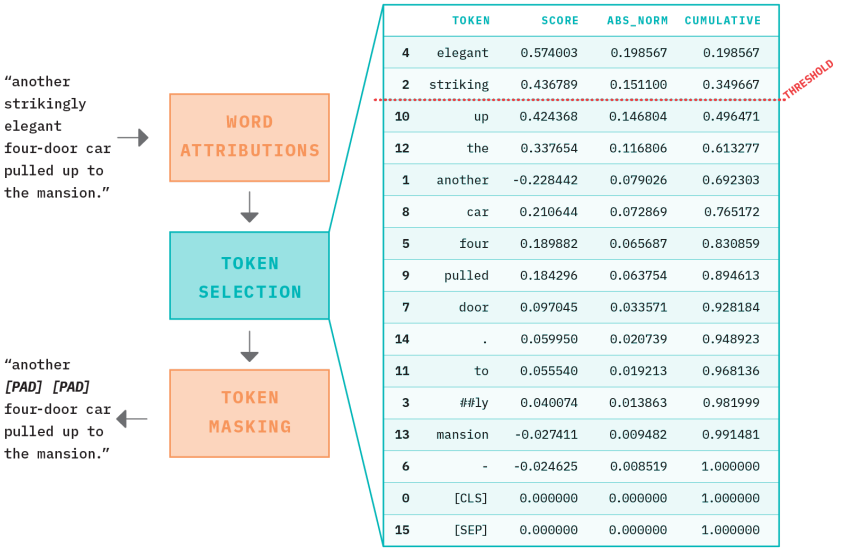


Figure 15: Word attributions visualized with Transformers Interpret for two sentences using integrated gradients on the fine-tuned BERT classification model. Positive attribution numbers (green) indicate a token contributes positively towards the predicted class ("subjective"), while negative numbers (red) indicate a word contributes negatively towards the predicted class.

The figure above demonstrates the power of contextual representations. In the first sentence (1), we see that the word "passing" is strongly attributed to the subjective classification of this sentence. That's because the term "passing" is a euphemism for "death"; a common NPOV-related correction in the WNC dataset. However, the term "passing" also appears in the second sentence (2), but is not attributed to the overall classification. That's because the BERT model recognizes that when used in this context, "passing" does not suggest death, but rather the act of physical movement, which is neutral in tone. Had we used a global style lexicon to replace subjective words, "passing" would have been erroneously removed from the second sentence.

Provided these token level attribution scores, we must then select which are considered stylistic elements to be masked out. To do so, we sort tokens in each sentence by the absolute, normalized attribution score and calculate a cumulative sum. This vector allows us to enforce a threshold on how much of the "total style" should be masked from the sentence without having to specify an explicit number of tokens.



	TOKEN	SCORE	ABS_NORM	CUMULATIVE
4	elegant	0.574003	0.198567	0.198567
2	striking	0.436789	0.151100	0.349667
10	up	0.424368	0.146804	0.496471
12	the	0.337654	0.116806	0.613277
1	another	-0.228442	0.079026	0.692303
8	car	0.210644	0.072869	0.765172
5	four	0.189882	0.065687	0.830859
9	pulled	0.184296	0.063754	0.894613
7	door	0.097045	0.033571	0.928184
14	.	0.059950	0.020739	0.948923
11	to	0.055540	0.019213	0.968136
3	##ly	0.040074	0.013863	0.981999
13	mansion	-0.027411	0.009482	0.991481
6	-	-0.024625	0.008519	1.000000
0	[CLS]	0.000000	0.000000	1.000000
15	[SEP]	0.000000	0.000000	1.000000

Figure 16: Style masking logic determines which tokens are considered style elements and therefore masked out from the sentence prior to calculating similarity measure.

In Figure 16 above, we see how cumulative attribution scores form the basis of style token selection. In this example, the terms "elegant" and "striking" combined account for ~35% of the style classification importance. This methodology allows us to set a tunable threshold whereby we mask out all tokens that contribute to the top X% of classification importance. To "mask out" style tokens, we simply replace them with either the informationless "[PAD]" token or remove them completely by deleting in-place.

The goal of this masking activity is to create "style-independent" versions of the original input and output sentences. These style-independent texts are then encoded using a generic, pre-trained SentenceBERT model to produce sentence level embeddings. SentenceBERT is a modified version of BERT that uses siamese and triplet network structures to derive semantically meaningful sentence representations that can be compared easily using cosine similarity (see the section *"To BERT or not to BERT"* from our report on Few-Shot Text Classification for more on SentenceBERT). We chose this embedding-based similarity method because it overcomes the limitations of strict string matching

methods (like BLEU) by comparing continuous representations rather than lexical tokens.

Figure 17 below summarizes the logical workflow used in our implementation of Content Preservation Score.

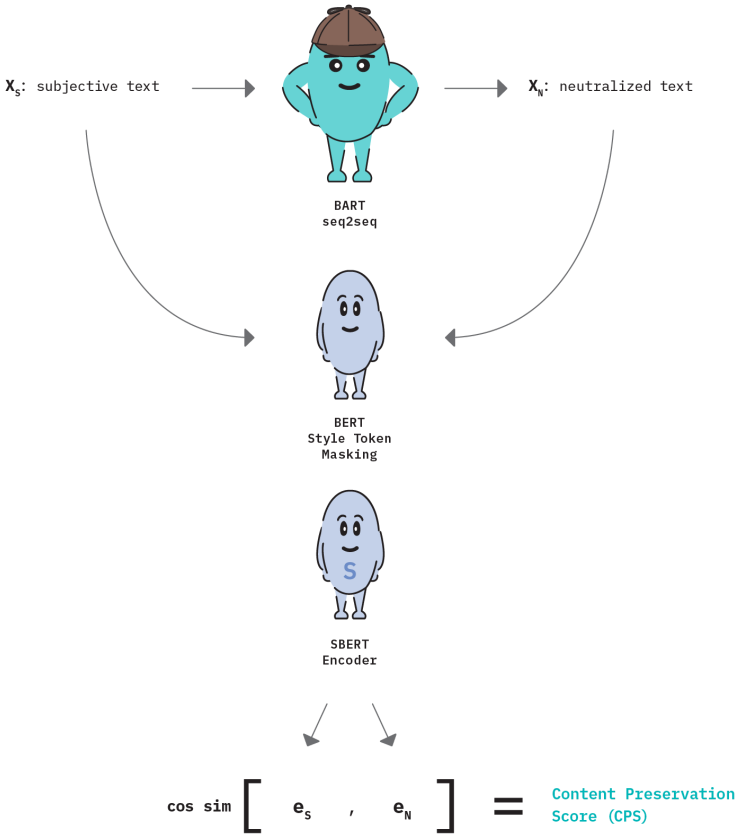


Figure 17: Content Preservation Score metric using BERT-based word attributions for style masking and SentenceBERT embeddings for similarity.

To begin (1), a fine-tuned text style transfer model (BART) is used to generate neutralized text *(X_N)* from a subjective input *(X_S)* .

Style tokens are then masked from both texts (2) using the methodology described in Figure 3 above to produce versions that contain only content-related tokens.

Next (3), the content-only texts are passed through a generic, pre-trained SentenceBERT model to produce a sentence embedding for each text (e_S , e_N).

Finally (4), we calculate cosine similarity between the embedding representations.

Considerations

While the high-level reasoning behind our implementations of STI and CPS make logical sense, there are nuances to the implementation that create room for error and jeopardize their effectiveness in measuring text style transfer. This is true of all automated metrics, and so we discuss these considerations below and recognize their importance as focus areas for future research.

Experimentally determining CPS parameters

To determine a default attribution threshold and masking strategy for our CPS metric, we experimentally searched over threshold values of 10% - 50% by 10% increments and masking strategies of “[PAD]” vs. removal while monitoring content preservation score on the held out test split. We also compare these parameter combinations to a case where no style masking is performed at all.

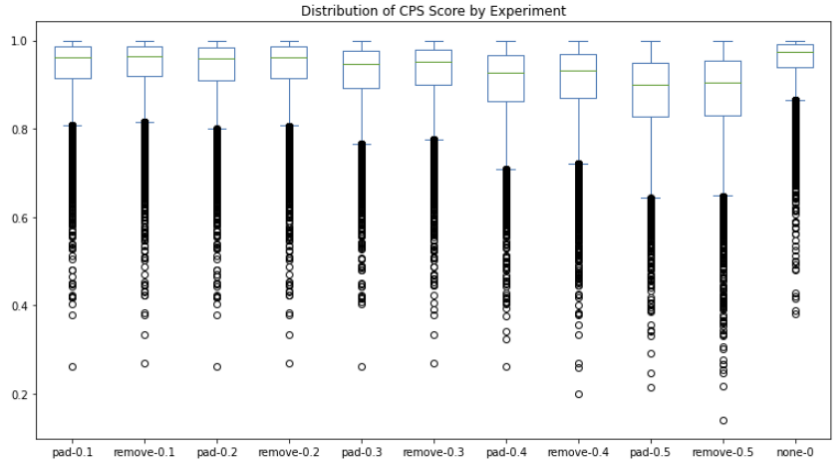


Figure 18: Content preservation score distributions across various experimental settings for *style threshold* and *masking strategy*.

We found that for each incremental threshold value, token removal produces a slightly higher average CPS score than replacement with "[PAD]" token. We also see that regardless of the parameter combination, all cases result in a lower median similarity score than had no style-masking been applied at all (see far right column in Figure 5). This makes sense because the more tokens we mask, the more opportunity there is to erroneously remove a piece of content instead of a stylistic element.

The only true way to determine the "best" parameter combination is to look at how CPS correlates with human evaluated scores. However, since we don't have access to manual evaluation scores, we select the combination that produces the highest outright CPS, which happens to be the case with no-style masking. For this reason, our CPS metric logic boils down to simply comparing SentenceBERT embeddings with cosine similarity, a similar landing place that others[21][22][23] have also arrived at.

Manual error analysis has revealed that our classifier-based attribution scores and style-token selection logic isn't consistent, nor precise enough at isolating only stylistic elements. As a result, meaningful content tokens are mistakenly removed which hinders content preservation measurement more than just leaving all tokens in. We discuss these challenges further in the following sections.

Dependence on style classifier

Both of the metrics we've implemented depend on the availability of a well-fit style classification model. This requirement translates to the need for labeled data. And while this isn't an issue for parallel TST tasks, it becomes a non-starter for the vast majority of style attributes where parallel data isn't available.

Even when parallel data is available, it's imperative that the trained model is performant. As we'll see in a later section, data quality issues can lead to a classifier that learns patterns that are unrepresentative of the true target attribute and result in an error-prone model. Since the STI metric is built directly on classifier distribution outputs, it is apparent that errors with the model will surface as errors in the style metric.

Similarly, the CPS metric uses the word attributions that are derived from the classifier as the basis for style token masking. If the model learns incorrect relationships between features and style classes, errors in word attribution scores can cause the wrong tokens to be masked. Unfortunately, there is minimal tolerance for error in style masking because incorrectly masking a content-related token (e.g. proper noun) can completely alter the semantics, producing a very low similarity score.

Style token selection

While our method for isolating style tokens in a sentence is driven by robust, contextual feature importances, the actual token selection methodology has room for improvement.

To begin, feature importances are attributed per token. Because BERT uses a word-piece tokenizer, we can see fragments of the same word with drastically different attribution values. For example, in Figure 3 above, we find that the term "strikingly" is tokenized in to the word pieces "striking" and "##ly", with the former attributed to ~15% importance and the latter just ~1%. Our current implementation considers these independently, and therefore applies a mask to just the root word alone. A considerable improvement would be to introduce logic that looks at combined scores for word pieces.

In addition, our method applies a "global" threshold to determine the amount of style (and therefore corresponding tokens) that are masked out. At a minimum, one token is always masked. This logic could be improved as there is likely a

relationship between length of sentence and maximum token feature importance. There are also cases where a sentence doesn't contain any style-related terms, and therefore masking one token incorrectly removes content by default.

Decoupling style from content

Our content preservation metric naively assumes that style can in fact be disentangled from content.

This remains an open question in the NLP research community[24], but from our experience and research, it seems there is growing consensus that style is woven into the content of written language, and the degree to which it can be separated is highly attribute-dependent. For the attribute of subjectivity, we believe that style is (at least partially) separable from content. For example, removing subjective modifiers (e.g. adjectives and adverbs) can change the style of a sentence without unwanted impact to semantics.

However, the challenge arises when theory meets practice. As we've found, automated methods for disentangling stylistic elements are consistently fraught with error, especially when operating in the lexical space (i.e. masking tokens). Newer approaches to text style transfer propose the separation of style from semantics in latent space representations with both supervised and unsupervised methods. We are encouraged by these efforts and look forward to continued research on this topic.

Are these metrics better than BLEU?

While we believe the STI and CPS metrics enable a more nuanced evaluation of text style transfer output than a singular BLEU score, we cannot say if these metrics are "better" without some human evaluated baseline to compare against. A "better" evaluation metric is just one that correlates stronger with human judgment, as this is the ultimate goal of automated evaluation.

Unfortunately, conducting human evaluation is outside the scope of our current research endeavor, but we do propose this as future work to build upon. In particular, we suggest conducting human evaluation in accord with this paper as a means to produce reliable evaluation benchmarks.

It's important to note that while human evaluation is the "best" means for evaluating generated text, it still isn't without issue. That's because determining if something is subjective vs. neutral is itself subjective. Subjective evaluation tasks likely lead to a higher degree of variability even among human reviewers.

Evaluating BART with STI & CPS

With our custom metrics defined, we utilize them to evaluate our fine-tuned BART model's ability to neutralize subjective language on the held out test set. We calculate STI and CPS scores between both the source and generated text, as well as the source and ground truth target annotation. Comparing metrics across these pairs helps build intuition for their overall usefulness and enables us to isolate edge cases of unexpected performance.

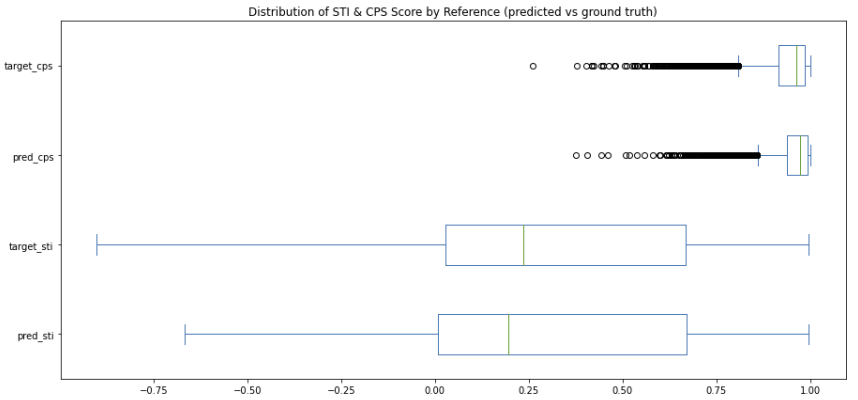


Figure 19: Distribution of STI and CPS scores on the held out test set. "Pred" corresponds to scores between source and generated text, while "target" corresponds to scores between source and ground truth annotation.

Content Preservation Score (CPS)

CPS score is built on cosine similarity and so it naturally ranges from 0-1. Figure 6 highlights the strong, left-skewed distribution of both the source-to-target and source-to-prediction examples. This result makes sense because we expect input and output pairs to be largely similar in semantics as that is the essence of this task and dataset – to slightly modify style while retaining meaning.

We see very similar distributions between target and predicted samples, with the source-to-predicted pairs having slightly higher median CPS scores and a smaller standard deviation. As we'll see, this finding hints at the conservative nature of our model (i.e. modest edits compared to human-made edits), as well as the perceptible data quality issues present in the full WNC corpus.

We analyze edge cases of mismatched performance between the model outputs and ground truths in Figure 7 below to better understand the strengths and weaknesses of our metrics.

#	Source Text	Target Text	Predicted Text	Target CPS	Pred CPS
1.	they have been the most successful of any english, spanish or italian club over the last 20 years, having won 18 major honours starting from the 1986-1987 season.	they have been the most successful english club over the last 20 years, having won 18 major honours starting from the 1986-1987 season.	they have won 18 major honours starting from the 1986-1987 season.	0.86	0.64
2.	however, this was last done during ww i during debates on denmarks neutral status.	however, this was last done during ww i in debates on denmark's neutral status .	however, this was last done during ww i during debates on neutral status.	0.99	0.79
3.	the most serious scandal was the iran-contra affair.	the best-known scandal was the iran-contra affair.	one controversy was the iran-contra affair.	0.94	0.73
4.	a series of events would bring this to the media's attention.	a series of events brought race relations on penn's campus to the media's attention.	a series of events would bring this to public attention.	0.48	0.86

Figure 20: Sample WNC pairs that demonstrate common themes around the CPS metric. Specifically, cases where target_cps >> predicted_cps (1-3) and target_cps << predicted_cps (4).

From Figure 20, we see that examples 1-3 highlight the scenario where the ground truth annotation preserves content much better (as defined by CPS) than the model's output, and the opposite for example 4. These examples demonstrate common themes (numerically matched below) that we've found through our error analysis.

1. ***The BART model tends toward brevity -*** The trained seq2seq model has learned that omission of content is generally a good tactic for reducing subjectivity. This is seen in the example above where the model selects an abbreviated version of the input. Because the model omits part of the content (i.e. "being the most successful club"), our CPS metric punishes the score relative to the ground truth.
2. ***SentenceBERT penalizes missing content -*** As expected, SentenceBERT embedding similarity captures the omission of important words. In this example, the prediction is penalized for dropping the important subject "Denmark".

3. ***CPS slips when style tokens are the difference -*** In contrast with example #2, our CPS metric struggles when the omitted words (“most serious”) are actually style related. In this example, CPS produces a disagreeably low score for the prediction as compared to the ground truth despite it largely retaining the semantic meaning. This demonstrates the imperative of isolating style elements from content. We tested removing these style-related terms (“most serious”) which resulted in a CPS score more representative of the semantic alignment.
4. ***Factual edits are out-of-scope -*** In this example, our model generated text produces a much higher CPS than the ground truth. This is due to the annotator’s introduction of new facts, or out-of-context information, that the model should not be expected to produce. We consider edits of this type to be outside the scope of our intended modeling task.

Overall, we see that our CPS metric has its strengths and weaknesses. We believe this metric is useful for providing a general indication of content preservation because low scores truly mark dissimilar content. However, this metric lacks marginal specificity and struggles to quantify small differences in content with accuracy.

Style Transfer Intensity (STI)

Unlike CPS, style transfer intensity ranges from -1 to 1 because movements away from the target style are penalized. We see from Figure 6 (above) and Figure 21 (below) that source-to-target and source-to-prediction STI distributions are very similar, which suggests the style transfer model is generally doing a good job of neutralizing text to resemble that of the ground truth.

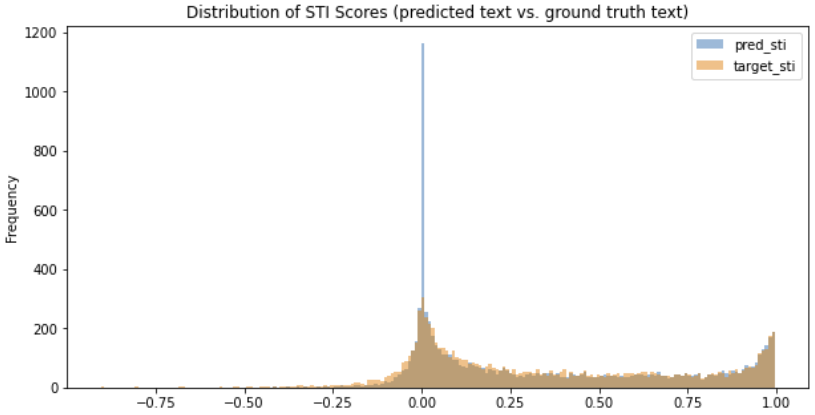


Figure 21: Histogram of STI scores on held out test set. “Pred” corresponds to scores between source and generated text, while “target” corresponds to scores between source and ground truth annotation.

However, there is a clear discrepancy between the distributions at STI value of 0. Here we see a significant number of generations that result in no change in style – these are cases where we found the model simply repeats the input as output. This implies that model is conservative in nature (i.e. refrains from making unsure edits) and explains the lower median STI score for the source-to-target population (0.19 vs. 0.24)

#	*Source Text*	*Target Text*	*Predicted Text*	*Target STI*	*Pred STI*
1.	wired equivalent privacy (wep) is an algorithm to secure ieee 802.11 wireless networks.	wired equivalent privacy (wep) is a flawed algorithm to secure ieee 802.11 wireless networks.	wired equivalent privacy (wep) is an algorithm intended to secure ieee 802.11 wireless networks.	**-0.81**	**0.08**
2.	building a difference received the prestigious seal of approval from the parents television council for the moving episode "page-bailie bunch".	building a difference received the seal of approval from the parents television council for the episode "page-bailie bunch".	building a difference received the seal of approval from the parents television council for the moving episode "page-bailie bunch".	**0.99**	**0.01**
3.	t. graham brown (born october 30, 1954, atlantageorgia), born anthony graham brown, is a grammy nominated (best roots gospel) american country/soul/gospel singer.	t. graham brown (born october 30, 1954, atlantageorgia), born anthony graham brown, is an american country/soul/gospel singer.	t. graham brown (born october 30, 1954, atlantageorgia), born anthony graham brown, is a grammy nominated american country/soul/gospel singer.	**0.98**	**-0.01**
4.	there is an iconic roadhouse, named "spud's roadhouse", which sells fuel and general shop items , has great meals and has accommodation.	there is an iconic roadhouse, named "spud's roadhouse", which sells fuel and general shop items and meals and has accommodation.	there is a roadhouse, named "spud's roadhouse", which sells fuel and general shop items and has accommodation.	**0.04**	**0.98**

Figure 22: Sample WNC pairs that demonstrate common themes around the STI metric. Specifcally, cases where target_sti < 0 (1), target_sti >> pred_sti (2-3), and target_sti << pred_sti (4).

Like in the CPS analysis, we can look at edge cases shown in Figure 22 to highlight themes about model and metric quality.

1. ***Incorrect target annotations -*** Figure 22 reveals that there are examples where the ground truth STI score is negative – implying that the ground truth annotation is more subjective than the source, which we can verify by looking at this first example. We see that the target text introduces the subjective modifier "flawed", which is clearly a labeling error. There are quite a few of these data quality issues that should be investigated and corrected in the dataset for future work.
2. ***BART can be partially correct -*** As shown here, there are many instances where the style transfer model correctly edits one instance of subjectivity in a sentence (e.g. removes "prestigious"), but misses additional occurrences (e.g. "moving").
3. ***Classifier error surfaces in STI metric -*** As discussed previously, STI depends on the quality of the style classification model. This example shows where the classifier incorrectly associates "grammy nominated" as a subjective modifier, when in fact the modifier phrase consists of neutral content.
4. ***BART sometimes does better than ground truth -*** By inspecting cases where target_sti << pred_sti, we find examples where the fine-tuned style transfer model legitimately outperforms the ground truth – a hopeful insight into the potential usefulness of the model.

Interpreting the STI metric

Style transfer intensity, as defined above, produces a directional magnitude indicating the distributional shift between style classifications from an input and output text. While this is a useful metric, it is difficult to compare across examples because the value is not normalized. For example, an STI score of 0.1 appears to be a weak indication of style transfer. But if that score corresponds to a distribution shift from [0.1, 0.9] to [0.0, 1.0], it actually represents the maximum possible shift in style because the distribution only had little room for improvement. Therefore, what appears to be a low STI score actually captured 100% of the possible target style gap. It would make little sense to put this example on the same footing as a distribution shift from [0.9, 0.1] to [0.8, 0.2].

This highlights the fact that STI should be measured relative to the total *potential* for style transfer. For this reason, we recommend representing STI as a percentage of the total possible, directionally corrected STI gain. If the output

text distribution moves closer towards the target style class, the metric represents the percentage of the possible *target* style distribution that was captured. If output text distribution moves further from the target style class, the metric represents the percentage of the possible *source* style distribution that was captured.

Ethical Considerations

In this final section, we'll discuss some ethical considerations when working with natural language generation systems and describe the design of our prototype application: Exploring Intelligent Writing Assistance.

Ethics as a criteria for topic selection in research

Standard practices for "responsible research" in the field of machine learning have begun to take hold. We now have datasheets for novel datasets, which are intended to document a dataset's motivation, composition, collection process, source of bias, and intended use[25]. Similarly, we have model cards that encourage transparent model reporting by detailing expected usage, performance characteristics, and model lineage[26]. While adoption of these practices still has room to grow, the seed is planted and has laid the foundation for increased transparency and accountability within the machine learning community.

However, both of these artifacts are backward looking – describing considerations of work products that have already been created. It is equally as important to consider ethical implications at the genesis of a project, before any research effort is underway. Similar to datasheets and model cards, *ethics sheets* have been proposed to encourage researchers to think about ethical considerations not just at the level of individual models and datasets, but also at the level of ML/AI tasks prior to engaging in a research endeavor[27]. An ethics sheet for an AI task is a semi-standardized article that aggregates and organizes a wide variety of ethical considerations relevant for that task. Creating an ethically focused document before researching or building an AI system opens discussion channels, creates accountability, and may even discourage project pursuance based on the supporting analysis.

For these reasons, our team engaged in brainstorming activity prior to researching the task of "automatically neutralizing subjectivity bias in text" to

consider potential benefits and harms of exploring and modeling the style attribute of subjectivity. We review some of our considerations below.

Potential benefits

As discussed in the earlier, subjective language is all around us. It makes for a useful style of communication by which we express ourselves and influence others. However, there are certain modes of communication today like textbooks and encyclopedias that strive for neutrality. A neutral tone is what this type of audience expects and demands.

In this context, a tool to automatically detect subjectively-toned language and suggest neutrally-toned counterparts could be helpful for several parties. For authors and editors, a tool of this kind could enable more efficient and comprehensive review of new and existing content – resulting in a higher standard of quality throughout published material. For content consumers, this type of tool could provide reading assistance to help alert readers when subjectivity bias is concealed within content they perceive to be neutrally-toned and factual.

Potential risks

Most modern language models used for generative tasks today build representations based on massive, uncensored datasets, which are subsequently fine-tuned on a smaller, focused corpora for a particular task. Therefore, these fine-tuned models inherit all of the potential risks associated with the large foundation models, plus any application specific concerns.

In this sense, our task adopts the risk of a model unintentionally reflecting unjust, toxic, and oppressive speech present in the training data. The consequences of this are that learning and projecting unknown biases can perpetuate social exclusion, discrimination, and hate speech[28]. Language models also risk introducing factually false, misleading, or sensitive information into generated outputs.

There is also the potential for malicious actors to intentionally cause harm with such a tool. While our efforts focus only on modeling the *subjective-to-neutral* style attribute direction, successful methods for generating neutral-toned text could be reverse engineered to model the opposite. Generating subjectively

biased text, automatically and at scale, could be used to undermine public discourse.

Similarly, adapting a successful modeling approach to a tangentially related style transfer task (e.g. political slant) could be used to exploit the [political] views of the masses if used for a malevolent social agenda. And finally, what is a world without opinion? A model that can silence the expressiveness of individual language could numb our ability to convey thoughts and feelings in online channels.

Should these risks discourage research

An upfront discussion of ethics is intended to capture various considerations that should be taken into account when deciding whether to develop a certain system, how it should be built, and how to assess its societal impact[29]. Ultimately, the concerns we've raised above do not simply "go away" by not exploring them. Instead, given the existing maturity of this field of NLP, we view this as an opportunity to increase transparency by surfacing the risks, along with our findings, best practices, and mitigating strategies.

Designing an intelligent writing assistant

To highlight the potential of this NLP task, we've bundled together our research artifacts into an *intelligent writing assistance* application that demonstrates how text style transfer can be used to enhance the human writing experience.

We emphasize the imperative for a human-in-the-loop user experience as a risk-mitigation strategy when designing natural language generation systems. We believe text style transfer has the potential to empower writers to better express themselves, but not by blindly generating text. Rather, generative models, in conjunction with interpretability methods, should be combined to help writers understand the nuances of linguistic style and suggest stylistic edits that *may* improve their writing.

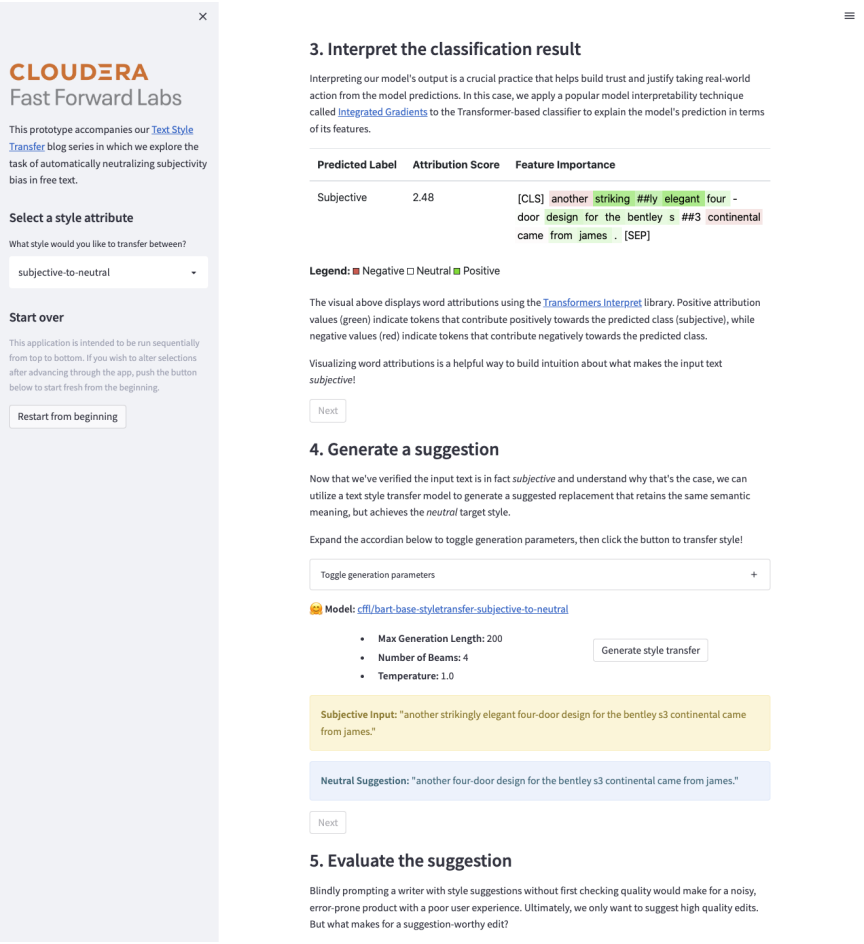


Figure 23: Screenshot of our prototype application: *Exploring Intelligent Writing Assistance*.

The goal of this application is to peel back the curtains on how an intelligent writing assistant might function — walking through the logical steps needed to automatically re-style a piece of text while building up confidence in the model output.

The user can choose to transfer style between two style attributes: *subjective-to-neutral* or *informal-to-formal*. After entering some text (or selecting a preset option), the input is classified to detect if a style transfer is actually needed. Then, an interpretability technique called Integrated Gradients is used to

explain the classifier's predictions in terms of its features, giving the user a look at what lexical components constitute a particular style. Next, the user can generate a style transfer while toggling the sequence-to-sequence model's decoding parameters. Finally, the generated suggestion is evaluated to provide the user with a measure of quality via two automated metrics: *Style Transfer Intensity (STI)* and *Content Preservation Score (CPS)*.

Conclusion

At last, we've made it to the final chapter of this research report. We started by broadly introducing the NLP task of text style transfer and discussing the often overlooked, but important role that style plays in the successful adoption of NLP technologies. We then explored how conditional language modeling approaches can be applied to the task of automatically neutralizing subjectivity bias. In doing so, we were faced with the nuanced difficulty of evaluating natural language generation (NLG), and implemented automated metrics to quantify style transfer strength and content preservation for our model outputs. Finally, we discussed some ethical considerations that should be attended to when designing an NLG system and described our prototype.

We hope you've enjoyed this report as much as we've enjoyed researching and writing about this exciting topic. We'll close out this series with a listing of all project outputs for quick reference.

Research Report:

- You're reading it!

Blog Series:

- Part 1: An Introduction to Text Style Transfer
- Part 2: Neutralizing Subjectivity Bias with HuggingFace Transformers
- Part 3: Automated Metrics for Evaluating Text Style Transfer
- Part 4: Ethical Considerations When Designing NLG Systems

Code:

- Research Code - Text Style Transfer: Neutralizing Subjectivity Bias with Huggingface Transformers
- Applied ML Prototype (AMP) - Exploring Intelligent Writing Assistance

HuggingFace Artifacts:

- Model: Subjective-neutral Style Classification

- Model: Subjective-to-neutral Style Transfer
- Space: Exploring Intelligent Writing Assistance

---

1. Politeness Transfer: A Tag and Generate Approach ↩
2. Deep Learning for Text Style Transfer: A Survey ↩
3. The Rest of the Story: Finding Meaning in Stylistic Variation ↩
4. Deep Learning for Text Style Transfer: A Survey ↩
5. Deep Learning for Text Style Transfer: A Survey ↩
6. Deep Learning for Text Style Transfer: A Survey ↩
7. Automatically Neutralizing Subjective Bias in Text ↩
8. Politeness Transfer: A Tag and Generate Approach ↩
9. Text Style Transfer: A Review and Experimental Evaluation ↩
10. Automatically Neutralizing Subjective Bias in Text ↩
11. Wikipedia:Neutral point of view ↩
12. Wikipedia: NPOV means neutral editing, not neutral content ↩
13. Wikipedia:Neutral point of view ↩
14. Linguistic Models for Analyzing and Detecting Biased Language ↩
15. Automatically Neutralizing Subjective Bias in Text ↩
16. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension ↩
17. BERTSCORE: Evaluating Text Generation with BERT ↩
18. Revisiting the Evaluation Metrics of Paraphrase Generation ↩

19. Evaluating Style Transfer for Text ↩

20. Evaluating Style Transfer for Text ↩

21. BERTSCORE: Evaluating Text Generation with BERT ↩

22. Style-transfer and Paraphrase: Looking for a Sensible Semantic Similarity Metric ↩

23. Deep Learning for Text Style Transfer: A Survey ↩

24. What is wrong with style transfer for texts? ↩

25. Datasheets for Datasets ↩

26. Model Cards for Model Reporting ↩

27. Ethics Sheets for AI Tasks ↩

28. Taxonomy of Risks posed by Language Models ↩

29. Ethics Sheets for AI Tasks ↩