Prompt-and-Rerank: A Method for Zero-Shot and Few-Shot Arbitrary Textual Style Transfer with Small Language Models

Mirac Suzgun[*]
Stanford University
msuzgun@cs.stanford.edu

Luke Melas-Kyriazi[*]
Oxford University
lukemk@robots.ox.ac.uk

Dan Jurafsky
Stanford University
jurafsky@cs.stanford.edu

Abstract

We propose a method for arbitrary textual style transfer (TST)—the task of transforming a text into any given style—utilizing general-purpose pre-trained language models. Our method, *Prompt-and-Rerank*, is based on a mathematical formulation of the TST task, decomposing it into three constituent components: *textual similarity*, *target style strength*, and *fluency*. Our method uses zero-shot or few-shot prompting to obtain a set of candidate generations in the target style, and then re-ranks them according to the three components. Our method enables small pre-trained language models to perform on par with state-of-the-art large-scale models while using two orders of magnitude less compute and memory. We also investigate the effect of model size and prompt design (e.g., prompt paraphrasing and delimiter-pair choice) on style transfer quality across seven diverse textual style transfer datasets, finding, among other things, that delimiter-pair choice has a large impact on performance, and that models have biases on the direction of style transfer.[1]

Sentiment Transfer	
Input	the food was low quality and thrown together quickly with little flavor.
GPT-J-6B	the food was delicious and prepared with a lot of care!
Input	the staff is awesome and location is right in the heart of old town!
GPT-J-6B	the staff is terrible and location is in the middle of nowhere!
Elizabethan → Modern English Translation	
Input	thou hast quar rell'd with a man for coughing in the street, because he hath wakened thy dog that hath lain asleep in the sun.
GPT-J-6B	you have cursed a man because he's coughed in the street, because he has woken your dog, which has been sleeping in the sun.
Input	if thou art fickle, what dost thou with him that is renown'd for faith?
GPT-J-6B	if you are fickle, then what are you doing with that person who is known for being reliable?
Fluency and Grammar Correction	
Input	Forexample, My cousin is 12years old.
GPT-J-6B	For example, My cousin is twelve years old.
Formality Transfer	
Input	r u talking about ur avatar?
GPT-J-6B	are you talking about your avatar?

Table 1: Qualitative examples of few-shot style transfer on the YELP, SHAKESPEARE, JFLEG, and GYAFC datasets. Coupling off-the-shelf "small" language models with our prompt-and-reranking method enables us to perform arbitrary textual style transfer without any model training or prompt-tuning. Compared to the extremely large language models (viz., ones with more than 100 billion parameters) used by Reif et al. (2022), our models obtain similar performance using almost two orders of magnitude less compute and memory.

1 Introduction

Textual style transfer (TST) refers to the task of transferring one stylistic aspect of a piece of text (e.g., sentiment polarity, formality, politeness, etc.) without changing its main semantic content, structure, or other attributes. Traditionally, the natural language generation (NLG) community has approached each instantiation of style transfer as a distinct task, designing and training specialized models on style-specific training corpora. For example, sentiment transfer has been studied extensively (Li et al. (2018); Sudhakar et al. (2019); Luo et al. (2019a), *inter alia*). This paradigm has restricted TST research to a limited, simple set of style choices with parallel corpora, which can often be solved adequately by word replacement (e.g., replacing negative words with corresponding positive words for sentiment transfer).

With the recent success of general-purpose language modeling (LM), it is, however, natural to ask whether one can tackle a more general formulation of style transfer: *arbitrary* TST, in which one aims to transform a reference text into an arbitrary style specified by the user at inference-time.

Inspired by the success of natural-language prompting in other domains (Radford et al., 2019; Petroni et al., 2019; Brown et al., 2020; Gao et al., 2021), we consider a prompting-based zero- and few-shot approach to arbitrary TST. Under this setup, we specify the desired type of style transfer problem using a natural-language prompt containing the source text (and optionally a few examples, in the few-shot case), and then use a pre-trained LM

[1]Our code, data, and results are available at https://github.com/suzgunmirac/prompt-and-rerank

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 2195 - 2222
December 7-11, 2022

to generate the stylized target text. Thus, the source text may be transformed into any user-specified style without additional training or fine-tuning.

Recent work (Reif et al., 2022) has found that extremely large language models (LLMs), namely the 175 billion-parameter GPT-3 (Brown et al., 2020) model and the proprietary 137 billion-parameter *LLM* model, are capable of sentiment and formality transfer. However, language models at this scale are not accessible to most researchers and practitioners, even in inference-only settings, due to their large memory consumption and slow generation times. Thus far, to the best of our knowledge, there has not been any research on the capabilities of reasonably-sized models for style transfer domain, nor any systematic study of how the precise construction of the prompt affects model performance.

Here we take a first-principles approach to arbitrary TST using pretrained language models. We first mathematically formalize the task, showing how it can be formulated as the combination of *textual similarity*, *target style strength*, and *fluency*. This framework naturally leads us to propose a new method for arbitrary TST, which we call "**Prompt-and-Rerank**." Using this method, we demonstrate, for the first time, that it is possible to perform arbitrary TST using reasonably-sized language models; prior work indicated that only enormous (i.e., GPT-3-scale) language models were capable of this task.

We summarize the main contributions and insights of this paper as follows: (i) We provide the first mathematical formalization of the arbitrary TST task. (ii) We propose Prompt-and-Rerank, a novel prompting-based method for arbitrary TST which follows naturally from our mathematical formulation. (iii) Our method matches and sometimes even exceeds state-of-the-art performance on arbitrary TST while using reasonably-sized language models such as GPT-2, which consume two orders of magnitude less memory and compute than prior work. (iv) We conduct a nuanced investigation of the influence of prompt design, such as task phrasing and delimiter-pair choice, on the quality of style transfer generations. (v) In order to encourage and facilitate further research in the area, we establish a set of benchmarks for arbitrary TST (including cleaned versions of the popular sentiment transfer datasets AMAZON and YELP) along with accompanying automatic evaluation metrics.

2 Background and Related Work

Background. TST is a long-standing problem in NLP which encompasses many popular sub-tasks, such as sentiment and formality transfer. Prior to the advent of large-scale pre-training in recent years, it was common practice to consider each of these sub-tasks separately, and to train separate models on different supervised datasets for each task. These models generally performed well within the limited scope of their task, but failed to generalize to new tasks or to texts outside of their training distribution. Here we show that the modern paradigm of pre-training large models and then prompting (or fine-tuning) them can be applied to many sub-tasks of TST in a unified, zero-shot manner, even with relatively small Transformers.

Related Work. Traditional approaches to TST can be broadly categorized into two families. The first family involves identifying and replacing distinctive style-related phrases (Li et al. (2018); Sudhakar et al. (2019); Wu et al. (2019); Madaan et al. (2020); Malmi et al. (2020); Reid and Zhong (2021), *inter alia*). For example, Madaan et al. (2020) perform the task of politeness transfer by first identifying words with stylistic attributes using TF-IDF and then training a model to replace or augment these stylistic words with ones associated with the target attribute. In general, these approaches perform well for very simple style edits (e.g., negating a sentence by adding the word *not*), but they struggle in scenarios that require more complex syntactic and semantic changes.

The second family of approaches involves disentangling latent representations of style and content, such that a text can be encoded into a style-invariant representation and then decoded in a desired style (Hu et al., 2017; Shen et al., 2017; Fu et al., 2018; Luo et al., 2019a). For example, Hu et al. (2017) encodes into and decodes from a style-agnostic latent space using a VAE alongside attribute discriminators. These approaches are often theoretically well-grounded, but they generally require large quantities of labeled data and struggle to scale beyond a small number of styles.

Differently from these two families, one recent work (Reif et al., 2022) uses enormous pre-trained language models to tackle TST, an idea motivated by the remarkable performance of pre-trained LMs in other areas of NLP (Radford et al., 2019; Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019). Specifically, they use *LLM*, *LLM-Dialog*, and GPT-

3, each of which has over 100 billion parameters, to rewrite texts in a variety of styles. However, they perform minimal analysis of their prompting setup, deferring such analysis to future work, and they suggest that this prompting-based approach is only feasible with LLMs.[2]

While drawing on many intuitions from Reif et al. (2022) and these earlier studies, this paper presents a novel prompt-and-rerank approach to the general task of TST using pre-trained language models. Alongside our method, we present the first systematic study of prompt formulation and model size for the task of textual style transfer. Contrary to expectations, using our method we find that even small LMs are able to effectively perform arbitrary style transfer. In fact, we match the performance of Reif et al. (2022) on multiple datasets using two orders of magnitude less memory and compute.

3 Method: Prompt-Based Arbitrary TST

This section begins with a mathematical formalization of the task of textual style transfer.[3] Our formalization elucidates the three underlying components of the task, namely *text similarity*, *target style strength*, and *fluency*, and naturally leads us to *Prompt-and-Rerank*, our prompt-based re-ranking algorithm for solving TST.

3.1 Problem Formulation

Let $\boldsymbol{x} \in \Sigma^*$ denote a text over a vocabulary Σ, and $\mathcal{S}$ the set of all possible text style choices. Let us further use $\boldsymbol{x}^{(s_1)} \in \Sigma^*$ to denote a text $\boldsymbol{x}$ written in the style $s_1 \in \mathcal{S}$. Informally speaking, the goal of TST is to transfer the style of a text $\boldsymbol{x}^{(s_1)}$ (usually, a sentence) from s_1 to s_2 without changing the main semantic content of the text. We can formally express this transformation via a function $f : \Sigma^* \times \mathcal{S} \times \mathcal{S} \rightarrow \Sigma^*$, which takes an input text (say $\boldsymbol{x}^{(s_1)}$) and its corresponding style (s_1), as well as a target style (s_2), and outputs a modified version of the input written in the style of s_2 (namely, $\tilde{\boldsymbol{x}}^{(s_2)}$).[4] Ideally, we would want the generated output $\tilde{\boldsymbol{x}}^{(s_2)} = f(\boldsymbol{x}^{(s_1)}, s_1, s_2)$ to be "close" (both semantically and syntactically) to the ground-truth $\boldsymbol{x}^{(s_2)}$ as much as possible.

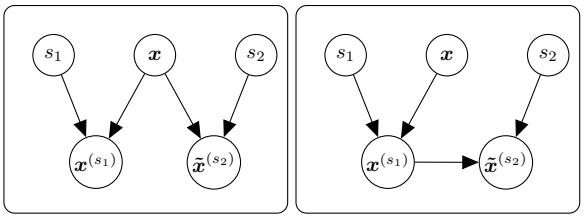

Figure 1: Two different but equally meaningful and valid interpretations of the textual style transfer task. Here $\boldsymbol{x}$ can be thought as the universal (abstract) meaning of a text, $\boldsymbol{x}^{(s_1)}$ a rewrite of $\boldsymbol{x}$ in the style of s_1. Depending on which graphical model one adheres to, $\boldsymbol{x}^{(s_2)}$ can be said to generated by $\boldsymbol{x}$ and s_2 (left model) or by $\boldsymbol{x}^{(s_1)}$ and s_2 (right model). In this paper, we follow the second interpretation.

The graphical models depicted in Figure 1 provide two different ways of formulating the task of TST (and of machine translation for that matter). Both models have valid and meaningful implications and interpretations; the main generative difference between them is that the parents of $\tilde{\boldsymbol{x}}^{(s_2)}$ are $\boldsymbol{x}$ and s_2 in the former (left), whereas the parents of $\tilde{\boldsymbol{x}}^{(s_2)}$ are $\boldsymbol{x}^{(s_1)}$ and s_2 in the latter (right).

Due to the inherent difficulty of collecting diverse supervised data for arbitrary TST, most prior studies considered a simplified version of the task, wherein the source (s_1) and target (s_2) style choices are fixed beforehand. In this work, we consider a broad formulation of the task, make no assumptions about the source and target style choices a priori, and explain how one can leverage the power of off-the-shelf LMs to perform arbitrary TST.

Given an input text $\boldsymbol{x}^{(s_1)}$ written in the style of s_1 and the target style s_2, we decompose the conditional likelihood of a generated output $\tilde{\boldsymbol{x}}^{(s_2)}$ into three terms:[5]

$$
\begin{aligned}
& p(\tilde{\boldsymbol{x}}^{(s_2)} \mid [\boldsymbol{x}^{(s_1)}, s_1], s_2) \qquad (1)\\
&= \frac{p(\tilde{\boldsymbol{x}}^{(s_2)}, [\boldsymbol{x}^{(s_1)}, s_1], s_2)}{p([\boldsymbol{x}^{(s_1)}, s_1], s_2)} \\
&\propto p([\boldsymbol{x}^{(s_1)}, s_1], [\tilde{\boldsymbol{x}}^{(s_2)}, s_2]) \\
&= p([\tilde{\boldsymbol{x}}^{(s_2)}, s_2])\; p([\boldsymbol{x}^{(s_1)}, s_1] \mid [\tilde{\boldsymbol{x}}^{(s_2)}, s_2]) \\
&= \underbrace{p(\tilde{\boldsymbol{x}}^{(s_2)})}_{\text{fluency}} \underbrace{p(s_2 \mid \tilde{\boldsymbol{x}}^{(s_2)})}_{\text{transfer strength}} \underbrace{p([\boldsymbol{x}^{(s_1)}, s_1] \mid [\tilde{\boldsymbol{x}}^{(s_2)}, s_2])}_{\text{textual similarity}}
\end{aligned}
$$

[2] A note on *terminology*: We shall refer to GPT-3 (Brown et al., 2020) and similar models with 100+ billion model parameters as *large* or *enormous* language models, as they are two-to-three orders of magnitude larger than previous models (e.g. the GPT-2 series with 117M-to-6B parameters).

[3] Despite its important role in NLG, we are not aware of any prior formal statement of the style transfer problem. Here, we hope to solidify the problem formulation and illustrate the a connection between this problem formulation and the automatic metrics used in the field to evaluate TST models.

[4] In cases where the original style of the input text might not be known a priori, one can either estimate the style of the input using a statistical classifier or assume that the input is written in a neutral style.

[5] We make use of the brackets "$[\cdot]$" only to group relevant terms (e.g., $\boldsymbol{x}^{(s_1)}, s_1$) together; they do not have any statistical significance in this context.

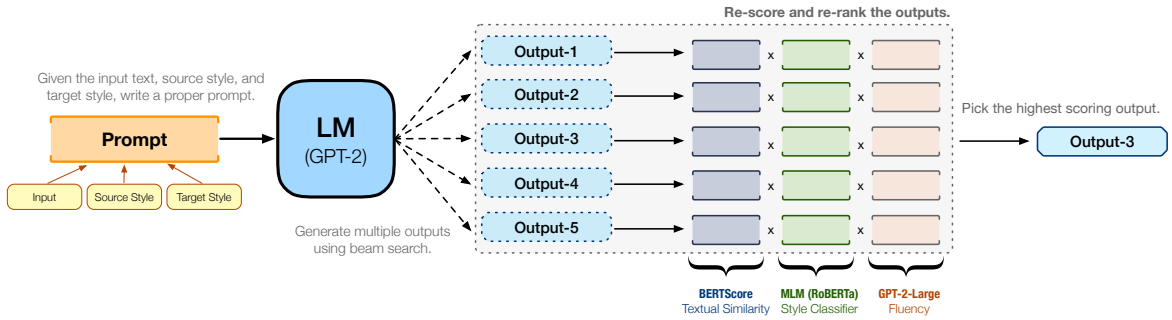

Figure 2: An illustration of our *Prompt-and-Rerank* method. Given an input text and a target style, we first compose a prompt and feed it to a pretrained language model to generate multiple output texts—conditioned on the prompt—using beam search. We then re-score each candidate output along the three axes from Eq. (1): *textual similarity*, *style transfer strength*, and *fluency*. We choose the candidate with the highest score as our final output.

The first term, $p(\tilde{\boldsymbol{x}}^{(s_2)})$, measures the overall *fluency* of the output. The second term, $p(s_2 \mid \tilde{\boldsymbol{x}}^{(s_2)})$, measures the *transfer strength* of the output (i.e., determines whether the output is written in the target style). The last term, $p([\boldsymbol{x}^{(s_1)}, s_1] \mid [\tilde{\boldsymbol{x}}^{(s_2)}, s_2])$, can be thought of as a proxy for *textual similarity* in the context of textual style transfer—it captures the correspondence between the input and output texts written in their respective styles.

3.2 *Prompt-and-Rerank* for Arbitrary TST

The problem formulation above naturally leads us to a method for (textual) style transfer, which we denote *Prompt-and-Rerank* (P&R).

The foundation of our method is use of *prompt templates* to convert TST into a natural-language generation problem. Formally, we use a predefined template $\tau \in \mathcal{T}$ to convert an input text $\boldsymbol{x}^{(s_1)}$ and the desired style transformation (i.e., $s_1 \rightarrow s_2$) into a natural-language prefix $\tau(\boldsymbol{x}, s_1, s_2)$. The template τ serves to not only contextualize the task for the model but also incorporate all the necessary conditional information (that is, input sentence, source style, and target style) in the input context. The precise design and composition of the templates is the topic of the following section (§4.2).[6]

Next, we feed the prompt into a pre-trained LM (e.g., GPT-2) and use the model to generate k different outputs $\tilde{\boldsymbol{x}}_k^{(s_2)}$ conditioned on the prompt, each sampled independently without updating any parameters of the model. These outputs are taken to be our *candidate* outputs for re-ranking. We then re-rank our k candidate outputs according to the decomposition in Equation 1:

$$p_{\text{reranking}}(\tilde{\boldsymbol{x}}_i^{(s_2)} | [\boldsymbol{x}^{(s_1)}, s_1], s_2) \propto p(\tilde{\boldsymbol{x}}_i^{(s_2)}) p(s_2 | \tilde{\boldsymbol{x}}_i^{(s_2)}) p([\boldsymbol{x}^{(s_1)}, s_1] | [\tilde{\boldsymbol{x}}_i^{(s_2)}, s_2]). \quad (2)$$

Finally, we pick the output $\tilde{\boldsymbol{x}}_i^{(s_2)} \in \tilde{\mathcal{X}}$ with the highest re-ranking score. Figure 2 provides an illustration of the method.

All that remains is to describe how to calculate each term in the re-ranking pass:

(i) To calculate the first (fluency) term, we use GPT-2-Large (774M) to determine the overall likelihood of each candidate text.[7]

(ii) For the second (transfer strength) term, we deliberately turn a *masked* language model (MLM), in our case a pre-trained RoBERTa model, into a *style* classifier as follows: Given $\tilde{\boldsymbol{x}}_i^{(s_2)} \in \tilde{\mathcal{X}}^{(s_2)}$ and $\mathcal{S} = \{s_1, s_2\}$, we convert $\tilde{\boldsymbol{x}}_i^{(s_2)}$ into a "fill-in-the-blank" cloze statement via a pre-defined cloze template, that is, we rewrite it as "The following text is <mask>: $[\tilde{\boldsymbol{x}}_i^{(s_2)}]$." We then query the MLM to predict the masked token,[8] but instead of looking at the probability distribution over the original model vocabulary, we restrict our attention to the elements in $\mathcal{S}$ and thus consider the likelihood of the missing token being s_1 or s_2. We then normalize these probabilities by l_1-normalization and get a proper probability distribution for $p(s_2|\tilde{\boldsymbol{x}}^{(s_2)})$.[9]

(iii) Finally, for the third (textual similarity) term, we use `BERTScore` (Zhang et al., 2020), which utilizes pre-trained contextual embeddings from BERT to measure the co-

[6] Additionally, in the few-shot case, where we have a number of few-shot exemplars, we convert these exemplars into meaningful prompts using the same template structure τ and prepend them to the main prompt.

[7] Given a text $x := x_{1:t}$ of length t, we calculate its probability under a model θ as $p_\theta(x) = \prod_{i=1}^{t} p_\theta(x_i \mid x_{<i})$

[8] One limitation of this framework is that it assumes the styles are associated with distinct tokens in the vocabulary.

[9] Of course, a more sophisticated normalization technique can be employed in this setup, but this basic normalization method seemed to be sufficient in our experiments.

sine similarity between two texts.[10] Specifically, we approximate $p([\boldsymbol{x}^{(s_1)}, s_1]|[\tilde{\boldsymbol{x}}^{(s_2)}, s_2])$ with $\texttt{BERTScore}(\boldsymbol{x}^{(s_1)}, \boldsymbol{x}^{(s_2)})$.[11]

Afterwards, we compute the score for each candidate by multiplying (i), (ii), and (iii) accordingly; re-rank all the candidates; and pick the one with the highest score as the final output.[12]

Overall, our approach is model-agnostic, allowing pre-trained LMs to be used out-of-the-box. Furthermore, our experiments show that with well-designed prompts, one does *not* need a massive language model for this approach to be successful.

4 Prompt Construction

In practice, we found the specific syntax and semantics of the prompt template significantly impact model performance. Thus, we conducted a systematic investigation of the impact of different prompt design choices on the quality of TST generations.

4.1 Delimiter-Pairs

We experimented with ten different text boundary markers (delimiter pairs), which may be divided into two categories: those whose opening and closing markers are identical (known as *indistinguishable* delimiters), and those whose markers are different (known as *complementary* delimiters). Specifically, we considered two indistinguishable pairs (viz., quotes and dashes) and eight complementary pairs: (1) curly brackets $\{\cdot\}$, (2) square brackets $[\cdot]$, (3) angle brackets $\langle\cdot\rangle$, (4) parentheses $(\cdot)$, (5) quotes " $\cdot$ ", (6) dashes – $\cdot$ –, (7) triple angle brackets $\langle\langle\langle\cdot\rangle\rangle\rangle$, (8) bracket quotes $\rangle$ " $\cdot$ ", (9) asterisk quotes * " $\cdot$ ", and (10) double curly bracket $\{\{\cdot\}\}$.[13] In their experiments, Reif et al. (2022) use only curly brackets.[14]

4.2 Prompt Phrasing

We considered four manually-written template formats $t \in \mathcal{T}$ for our discrete prompts:

(a) *Vanilla*: "Here is a text: $[d_1][\boldsymbol{x}^{(s_1)}][d_2]$ Here is a rewrite of the text, which is $[s_2]$: $[d_1]$",

(b) *Contrastive*: "Here is a text, which is $[s_1]$: $[d_1][\boldsymbol{x}^{(s_1)}][d_2]$ Here is a rewrite of the text, which is $[s_2]$: $[d_1]$",

(c) *Negation-v1*: "Here is a text, which is $[s_1]$: $[d_1][\boldsymbol{x}^{(s_1)}][d_2]$ Here is a rewrite of the text, which is not $[s_1]$: $[d_1]$", and

(d) *Negation-v2*: "Here is a text, which is not $[s_2]$: $[d_1][\boldsymbol{x}^{(s_1)}][d_2]$ Here is a rewrite of the text, which is $[s_2]$: $[d_1]$".

Note that $[d_1]$ and $[d_2]$ denote the opening and closing elements of the chosen delimiter-pair, respectively. In their experiments, Reif et al. (2022) exclusively made use of the *vanilla* setting, which only specifies the target style (s_2) in the second half of the prompt; however, we initially speculated that providing useful information about the source style (s_1) and creating a clear contrast between the source and target styles in the prompt semantics might help pre-trained LMs to have a better understanding of the underlying nature of the task and improve their performance; hence, we decided to look at the *contrastive* setting as well. As for the other two *negation* templates, we wanted to test how specifying the source style as the negation of the target style (viz., $s1$:="not s_2") and vice versa might affect the model performance.[15]

Example. Finally, to make our prompting setup more concrete, let us give a concrete and brief example of how we formulate a prompt. We consider the *contrastive* template with *curly brackets* as our delimiter. If we have an input sentence $\boldsymbol{x}^{(s_1)}$="I love *The Sound of Music*; it is the best movie ever!!" with s_1=*positive* and s_2=*negative*, then the prompt under this template would be "Here is a text, which is positive: {I love *The Sound of Music*; it is the best movie ever!!} Here is a rewrite of the text, which is negative: {" The language model would then generate an output by autoregressively decoding after the last delimiter, to produce a sentence such as: "I hate *The Sound of Music*; it is the worst movie ever!!}."[16]

[10] Our choice of BERTScore comes with some approximations. It is symmetric, i.e., $\texttt{BERTScore}(\boldsymbol{x}^{(s_1)}, \boldsymbol{x}^{(s_2)}) = \texttt{BERTScore}(\boldsymbol{x}^{(s_2)}, \boldsymbol{x}^{(s_1)})$, and BERTScore also does not explicitly include style information. We believe these are reasonable simplifications (and it is possible that pre-trained BERT implicitly incorporates style information).

[11] One can alternatively use MoverScore (Zhao et al., 2019), BARTScore (Yuan et al., 2021), or BLEURT (Sellam et al., 2020) for approximating textual similarity in (iii).

[12] Since the calculation of (iii) penalizes long sequences or sequences involving rare words, we also consider the re-ranking method in which we ignore the *fluency* factor, assuming that the sentences generated by the models are always fluent, which, we are aware that, is a faulty assumption.

[13] We use (8), (9), and (10) to emulate blockquotes, bullet points, and liquid tags in Markdown, respectively.

[14] We hypothesized that the complementary delimiter-pairs might yield better results than the indistinguishable ones, since it is categorically easier for models to distinguish and understand where sentences start and end. We also speculated that delimiter-pairs that were more likely to be used as text-separators in the training data in various contexts (e.g., in code snippets) might yield better results.

[15] The last two formats might be useful especially when we do not have access to either the source or the target style.

[16] Table 9 in the Appendix provides a complete set of exam-

Dataset	Styles	Example Sentence-Pairs	Test Set Size
Yelp Restaurant Reviews (Zhang et al., 2015)	Negative Positive	ever since joes has changed hands it's just gotten worse and worse. ever since joes has changed hands it's gotten better and better.	1000
Amazon Product Reviews (He and McAuley, 2016)	Negative Positive	if your bike had a kickstand on the plate it won't lock down. if your bike had a kickstand on the plate it would lock down.	1000
GYAFC Formality Dataset (Rao and Tetreault, 2018)	Informal Formal	and so what if it is a rebound relationship for both of you? what if it is a rebound relationship for both of you?	1000
Shakespearean English Dataset (Xu et al., 2012)	Elizabethan Modern	is rosaline, whom thou didst love so dear, so soon forsaken? have you given up so quickly on rosaline, whom you loved so much?	599
JFLEG Corpus (Napoles et al., 2017)	Ungrammatical Grammatical	Forexample, My cousin is 12years old. For example, my cousin is 12 years old.	747
Symbolic Manipulation (*Ours*)	Symbolic English	olive > cat olive is greater than cat	1000

Table 2: Overview of the textual style transfer datasets used in this paper.

4.3 Zero-Shot vs. Few-Shot Settings

In recent years, LLMs, such as GPT-3, have proven themselves to be resourceful few-shot learners. In a few-shot learning setting, a model is presented with a small set of illustrative examples, oftentimes along with a natural-language prompt describing the task, and expected to understand the underlying task and make accurate predictions without performing any gradient updates to the weights of the model at inference time. We wanted to explore how the number of demonstrations affects the performance of our models. To that end, we also tested the performances of our models under the zero-shot and four-shot settings.

5 Experiments and Results

5.1 Datasets

Differently from most previous work, which focused on single TST subtasks or datasets, we present experiments on a wide range of TST subtasks (also described in Table 2):

- **YELP**: Sentiment transfer for Yelp reviews (Zhang et al., 2015).
- **AMAZON**: Sentiment transfer for Amazon reviews (Li et al., 2018).
- **SHAKESPEARE**: *Elizabethan*-to-modern translation for Shakespeare (Xu et al., 2012).
- **GYAFC**: Formality transfer for Yahoo Answers responses (Rao and Tetreault, 2018).
- **JFLEG**: Grammar error correction for student essays (Napoles et al., 2017).
- **SYM**: Symbol-to-natural-language translation on a new custom synthetic dataset.

In the initial stages of our research, we noticed that all of these datasets, with the exception of SYM (which is synthetic), contain various tokenization issues (e.g., sentences sometimes contain extra white-space or have their punctuation marks separated out by spaces). We did not wish these tokenization artifacts to diminish the quality of our generations from general-purpose LMs—neither did we want this issue to negatively impact our evaluation scheme. To that end, we used a simple text-cleaning procedure to clean the texts.[17]

5.2 Evaluation Metrics

Prior studies on style and sentiment transfer have typically evaluated models across three dimensions: content/meaning preservation (textual similarity), style transfer strength, and fluency (Mir et al., 2019; Briakou et al., 2021). We note that these dimensions correspond exactly to the criteria that appear in Equation 1 in §3.1 (Krishna et al., 2020).

Content Preservation. BLEU (Papineni et al., 2002) is the standard metric for measuring semantic content preservation. We use the SacreBLEU (sBLEU) implementation (Post, 2018) to compute both *reference*-BLEU (*r*-sBLEU) and *self*-sBLEU (*s*-sBLEU) scores. Whereas *r*-sBLEU helps measure the distance of generated sentences from the ground-truth references, *s*-sBLEU indicates the degree to which the model directly copies the source.

Transfer Strength. In order to determine whether outputs generated by a TST model have the attributes of their target styles, we follow the standard classifier-based approach: we train a (binary) style classifier on the corpus of interest and use it to estimate the fraction of generated outputs whose styles match their target styles.

Fluency. To measure the fluency of generated texts, we compute their average token-level perplexity (PPL) using a pre-trained LM (in our case,

ples of prompts used in each task.

[17] We release both the original and cleaned versions of the datasets alongside this paper to help facilitate future research. In the Appendix, we also present results for both the original and cleaned datasets.

Model	Acc	*r*-sBLEU	*s*-sBLEU	PPL
SUPERVISED				
[1] CrossAlignment	0.73	7.8	18.3	217
[2] BackTrans	0.95	2.0	46.5	158
[3] MultiDecoder	0.46	13.0	39.4	373
[4] DeleteOnly	0.85	13.4	33.9	182
[4] DeleteAndRetrieve	0.90	14.7	36.4	180
[5] UnpairedRL	0.49	16.8	45.7	385
[6] DualRL	0.88	25.9	58.9	133
[7] ST (Multi-Class)	0.86	26.4	63.0	175
[7] ST (Conditional)	0.93	22.9	52.8	223
[8] B-GST	0.81	21.6	46.5	158
ZERO- OR FEW-SHOT INFERENCE ONLY				
[9] LLM $_{\text{Aug-0S-FirstChoice}}$	0.85	5.3	9.2	33
[9] LLM $_{\text{5S-FirstChoice}}$	0.93	6.7	11.2	43
[9] LLM $_{\text{Aug-0S-Best-sBLEU}}$ †	0.63	19.8	45.1	55
[9] LLM $_{\text{5S-Best-sBLEU}}$ †	0.78	23.2	48.3	77
Ours (GPT-2-XL)	0.87	14.8	28.7	65
Ours (GPT-J-6B)	0.87	23.0	47.7	80

Table 3: A comparison of our *Prompt-and-Rerank* approach with supervised sentiment transfer methods and the ultra-large-scale prompting-based method of Reif et al. (2022) on YELP-clean. In order to compare fairly against previous studies, we applied our data-cleaning code to their publicly-available outputs and re-computed all evaluation metrics. References: [1] (Shen et al., 2017), [2] (Prabhumoye et al., 2018), [3] (Fu et al., 2018), [4] (Li et al., 2018), [5] (Xu et al., 2018), [6] (Luo et al., 2019b), [7] (Dai et al., 2019), [8] (Sudhakar et al., 2019), [9] (Reif et al., 2022). Note on †: We used sBLEU to choose the best candidate, as opposed to BLEU that was used originally by Reif et al. (2022).

Dataset	Model	Acc*	*r*-sBLEU	*s*-sBLEU	PPL
AMAZON *P*→*N*	GPT-2-XL	**0.70**	11.5	17.2	77
	GPT-J-6B	0.65	**21.5**	31.4	**70**
AMAZON *N*→*P*	GPT-2-XL	**0.56**	13.2	19.9	**50**
	GPT-J-6B	0.52	**19.3**	29.3	58
YELP *P*→*N*	GPT-2-XL	**0.87**	14.8	28.7	**65**
	GPT-J-6B	**0.87**	**23.0**	47.7	80
YELP *N*→*P*	GPT-2-XL	**0.72**	12.0	25.3	**55**
	GPT-J-6B	0.65	**20.2**	44.6	58
SHAKE-SPEARE	GPT-2-XL	0.39	18.9	38.4	90
	GPT-J-6B	**0.78**	**21.9**	31.8	**81**
JFLEG	GPT-2-XL	35.9	**74.8**	91.5	76
	GPT-J-6B	**40.0**	64.8	59.1	**48**
GYAFC	GPT-2-XL	0.82	32.7	41.9	**58**
	GPT-Neo-1.3B	**0.85**	**36.4**	49.6	68
SYM	GPT-2-XL	0.56	68.5	-	-
	GPT-J-6B	**0.74**	**81.9**	-	-

Table 4: Four-shot performances of GPT-2-XL and GPT-J across all style transfer tasks, using curly brackets as delimiters. Full results with all models and delimiter pairs are shown in the appendix. *P*→*N* stands for the positive → negative direction, and vice-versa for *N*→*P*. * for JFLEG, GLEU score is used in place of accuracy to measure grammar correction performance. Note that the *r*-sBLEU column is not bolded because it is not necessarily desirable to have a higher *r*-sBLEU.

GPT-2-Large). We note that, whilst this PPL-driven approach has the advantage of being automated and practical, it still contains considerable drawbacks, including biases towards shorter texts.

5.3 Model Choices.

We used four GPT-2 models (Radford et al., 2019) of varying sizes (viz., GPT-2-Small (117M params), GPT-2-Medium (345M), GPT-2-Large (774M), and GPT-2-XL (1.6B)), GPT-Neo-1.3B Black et al. (2021), GPT-Neo-2.7B, and GPT-J-6B (Wang and Komatsuzaki, 2021). We highlight that none of these models were fine-tuned or prompt-tuned.

5.4 Results

Here, we present a summary of our key findings. For our complete results, we encourage the reader to see the Appendix (especially, Tables 11-20).

Table 3 juxtaposes our results on YELP with those of prior studies. Despite not training or fine-tuning, our method is competitive with prior models that were designed and trained specifically for these tasks. In fact, compared to supervised methods, our models almost always generate more fluent outputs, as measured by perplexity. Compared to Reif et al. (2022), who utilize the proprietary 137-billion-parameter *LLM* (LaMDA), we compare on-par or favorably despite using much smaller models; we obtain better sBLEU scores than their "FirstChoice" setting (which uses a single output) and better accuracy scores than their "BestBLEU" oracle setting (which takes the best of 16 outputs, as measured by sBLEU score).

Table 4 presents a summary of our results across all seven TST datasets for GPT-2-XL and GPT-J. For full results including all models (GPT-2-Small to GPT-J), please refer to the Appendix. Broadly, we find that all models are capable of TST to a reasonable degree—with the larger models (e.g., GPT-2-XL, GPT-Neo-2.7B, GPT-J) often performing better than the smaller models. The only model that consistently performs poorly is GPT-2-Small: Its high *s*-sBLEU and low accuracy indicate that it *copies* long sections of the input (without changing its style) more often than the other models.

Looking at individual tasks, we recognize that there remains substantial room for improvement on the JFLEG task: Most models underperformed a simple baseline that copied the input text without making any changes. The baseline achieved 37.2 GLEU, better than all models except GPT-J (which obtained 40.0). Finally, on our new synthetic task SYM, we found that GPT-J performed significantly

Setting	Target Style	Textual Similarity	Fluency
Ground Truth	3.60 (0.07)	3.34 (0.09)	3.78 (0.05)
DeleteAndRetrieve (Li et al., 2018)	2.31 (0.14)	2.10 (0.13)	1.97 (0.17)
Style Transformer (Dai et al., 2019)	2.43 (0.17)	2.94 (0.13)	2.18 (0.18)
LLM (Reif et al., 2022)	2.90 (0.13)	1.98 (0.14)	**3.73** (0.05)
Ours (*Prompt-and-Rerank*)	**3.32** (0.13)	**3.51** (0.10)	3.67 (0.08)

Table 5: Human evaluation results on YELP-clean. Each score in the table represents the mean of 250 ratings (with standard errors shown in parentheses): 5 ratings per example across 50 examples, of which 25 were positive-to-negative and 25 were negative-to-positive. Raters scored examples on a scale of 1-to-5 (with 5 being best) across three dimensions: target style, textual similarity, and fluency. We find that *Prompt-and-Rerank* performs well across all dimensions; it scores highest on *Target Style* and *Textual Similarity* and scores only slightly lower than LLM (a 137-billion parameter language model) on *Fluency*.

Setting	Acc	*r*-sBLEU	*s*-sBLEU	PPL
Vanilla	78.0	14.7	31.0	**58.5**
Contrastive	**79.5**	13.4	27.0	59.5
Negation-v1	66.5	13.4	28.1	67.5
Negation-v2	52.0	**18.0**	40.6	69.0

Table 6: Four-shot performances of GPT-2-XL on YELP-clean under different prompting protocols. We show the average of scores from $P \rightarrow N$ and $N \rightarrow P$ directions. A full table with all models is in the Appendix. Across all models, the vanilla and contrastive prompting protocols tend to yield the most favourable results.

Delimiter	Acc	*r*-sBLEU	*s*-sBLEU	PPL
⟨·⟩	49.5	17.4	40.8	45
* " · "	55.0	12.0	29.8	37
⟩ " · "	53.0	10.7	25.4	35
{·}	59.5	10.0	23.6	35
– · –	54.5	6.4	16.5	**24**
{{·}}	50.5	**18.3**	43.9	65
(·)	55.5	12.4	28.1	43
" · "	**60.5**	8.8	20.4	31
[·]	58.0	11.4	27.4	41

Table 7: Zero-shot performances of GPT-2-XL on YELP-clean using different delimiter pairs. Full tables with all models for all datasets are in the Appendix.

better than the rest: It achieved 74% accuracy[18] whereas no other model exceeded 60% accuracy.[19]

5.5 Human Evaluation

To evaluate the efficacy of our proposed Prompt-and-Rerank method, we also conducted a human-subject study. Our goals were (1) to assess how our proposed method fares against the previously proposed methods for style transfer, and (2) to understand how correct and well-written both the generations and ground-truth references are.

Our human evaluation followed the procedure from Reif et al. (2022), in which six human-raters were to asked to rank outputs along three dimensions—namely target-style strength, textual similarity, and fluency. Our six (volunteer) raters were all graduate students who were all native or fluent English speakers. For our evaluation, we focused only on YELP: We randomly sampled 50 examples (25 positive-to-negative and 25 negative-to-positive) and selected the corresponding outputs from DeleteAndRetrieve (Li et al., 2018), Style Transformer (Dai et al., 2019), *LLM* (Reif et al., 2022), and our own method "Prompt-and-Rerank", along with the ground-truth references. Each example (and its corresponding set of outputs) was rated by five raters.

To ensure fair comparison, we presented the same samples to our human-raters but randomized the order of the outputs and references, and we asked our raters to rate each output on a scale of 1-to-5, where 5 indicates the best and 1 the worst. All our participants successfully completed our study, and it took about one hour for each rater to complete our human evaluation study.

Results. Table 5 includes a summary of our human-evaluation results. "Prompt-and-Rerank" obtained the highest scores along both the target style and textual similarity dimensions, performing significantly better than the previously proposed methods. In terms of fluency, Reif et al. (2022)'s method (3.73) has scored slightly higher than ours (3.67), and both methods were close to the average ground-truth fluency score (3.78). We further note that the low textual (semantic) ranking score of the ground-truth references suggest that the references might be slightly noisy.

5.6 Further Analysis and Discussion

Contrastive prompting generally improves style transfer quality. As shown in Table 6 (and Table 20 in the Appendix), amongst the four prompt-

[18] Accuracy is measured via exact-string-matching.

[19] When the models failed to generate the correct output, we found that a common failure case was copying the input words correctly but using the wrong logic (e.g., generating "less than" instead of "greater than").

ing protocols considered in this paper, contrastive prompting generally yielded the best accuracy, albeit not always the best sBLEU scores.

Delimiter-pair choice has a large impact on model performance. Our systematic analysis of ten different delimiter-pairs shows that delimiter choice substantially affects the quality of generated outputs. Although there is not a single pair which performs best across all setups, certain delimiters, such as the curly brackets $\{\cdot\}$, square brackets $[\cdot]$, parentheses $(\cdot)$, and quotes " $\cdot$ ", yielded consistently better results on both AMAZON and YELP (see Tables 10-13). We hypothesize that the strong performance of these markers is attributable to the fact that they are often used as text separators (or dividers) in different textual contexts, such as essays, dialogues, and code snippets, which compose part of the pre-training data of our models.

Re-ranking improves overall performance. We considered two re-ranking approaches, one in which we picked the generated output with the highest beam score and one in which we sampled three outputs from the model using beam search and then re-scored them according to three criteria discussed in §3.2. As shown in Tables 15 and 16, the re-ranking method can boost the sentiment accuracy by 10-30%. It often, but not always, leads to better sBLEU and fluency scores. Also, as Table 8 illustrates, if we have access to a classifier trained on paired data, it might be more convenient to use it in our style transfer accuracy measurements, instead of an MLM as a proxy-classifier, in the re-ranking process, as it empirically leads to higher accuracy and sBLEU scores.

Analysis of bias and transfer performance in opposite directions. We find that pre-trained models have strong directional biases: None of the models performed the same when going in the negative→positive (N→P) and positive→negative (P→N) directions on AMAZON and YELP. We offer three possible explanations for this phenomenon: (i) The inherent differences in the linguistic difficulty of the tasks, (ii) the potential biases in pre-training dataset(s), and (iii) the poor quality of annotations in certain style transfer directions. Regarding (i), a qualitative inspection of the sentiment transfer datasets illustrates that in some cases, good P→N performance can be achieved by simply adding a negation (e.g., "not") into the text. Regarding (ii), it is possible that the web-scraped pre-training data of these models contains more sentences that resemble the task of changing the sentiment from positive to negative than the reverse direction during their pre-training periods. Qualitatively, the GPT-2 models appear adept at negation; therefore, it may not be surprising that these models yield better results in the P→N direction. As for (iii), our inspection of the ground-truth data reveals that it contains some noisy labels and incorrect input-output pairs.

Model	Setting	Acc	*r*-sBLEU	*s*-sBLEU	PPL
GPT-2-XL (1558M)	Top Choice	0.63	13.7	20.3	65
	P&R$_{\text{RoBERTa}}$	0.87	14.8	28.7	65
	P&R$_{\text{Oracle Cl.}}$	0.95	16.8	33.4	63
GPT-J-6B (6B)	Top Choice	0.81	25.3	50.5	107
	P&R$_{\text{RoBERTa}}$	0.87	23.0	47.7	80
	P&R$_{\text{Oracle Cl.}}$	0.95	25.4	52.4	87

Table 8: Comparison of vanilla four-shot performance of GPT-2 XL and GPT-J-6B models on YELP-clean ($P \rightarrow N$) under three settings: (1) choosing the output with the highest beam score (TC), (2) Prompt-and-Rerank with RoBERTa used as a zero-shot style classifier (P&R$_{\text{RoBERTa}}$), and (3) Prompt-and-Rerank with an oracle style classifier trained on paired data (P&R$_{\text{Oracle}}$). For full results in YELP-clean, see Table 16.

Limitations. The primary limitation of our re-ranking method is that it involves generating multiple outputs from an autoregressive LM, which requires multiple forward passes. Additionally, our approach relies on having access to a pre-trained bi-directional MLM. Compared to a simple zero-shot approach, these elements could potentially add complexity to deploying this model in practice.

6 Conclusion

In this paper, we propose a novel formal framework for textual style transfer. This framework naturally leads us to a new method, which we denote *Prompt-and-Rerank*, that utilizes general-purpose pretrained language models to transform text into in arbitrary styles. In our experiments, we use our method to demonstrate that off-the-shelf, pre-trained "small" language models, such as GPT-2, can perform arbitrary textual style transfer, without any additional model fine-tuning or prompt-tuning. Additionally, we conduct an extensive investigation prompt phrasing and delimiter choice on transfer quality. In total, we hope that our work makes further research in this area more accessible to a broad set of researchers, both by alleviating the computational constraints of hundred-billion-parameter language models and by establishing a standard set of clean datasets for arbitrary text style transfer.

7 Ethical Considerations & Limitations

Our work aims to advance the state of research on the task of arbitrary textual style transfer. As with many NLP applications, these methods may be used for negative purposes by malicious actors. For example, it would be possible to conceive of an instantiation of arbitrary textual style transfer which converts a non-sensationalist news headline into a sensationalist news headline, or one that converts a non-offensive piece of text into an offensive piece of text, in order to achieve a malicious goal.

Our work also involves pretrained general-purpose language models, which bring up less-obvious ethical considerations than those discussed above. Since these language models are trained on text scraped from the web, they have acquired some of the biases present in web text. Such biases may be extracted by certain forms of prompting; recent work (Prabhumoye et al., 2021) suggests that few-shot prompts can be used to detect social biases in pretrained language models. A large body of work is dedicated to understanding and de-biasing these large language models, but it is not the subject of our present work.

Acknowledgments

We would like to thank Yonatan Belinkov, Federico Bianchi, Dora Demszky, Esin Durmus, Tim Franzmeyer, Tayfun Gür, Tatsu Hashimoto, John Hewitt, Laurynas Karazija, Deniz Keleş, Faisal Ladhak, Percy Liang, Nelson Liu, Shikhar Murty, Tolúlọpẹ́ Ògúnrèmí, Isabel Papadimitriou, Ashwin Paranjape, Stuart M. Shieber, Suny Shtedritski, Kyle Swanson, Rose Wang, Elliot Wu, Tianyi Zhang, and Kaitlyn Zhou for their help and support. We especially thank Suproteem K. Sarkar for proofreading an earlier version of this manuscript and providing us with constructive comments and valuable suggestions. We also wish to thank the anonymous reviewers for providing us with helpful feedback, Sudha Rao for helping us with the navigation of the GYAFC dataset, and Yunli Wang for sharing their data splits and classifiers for GYAFC that were used in their paper. We gratefully acknowledge the support of Open Philanthropy, the NSF (via award IIS-2128145), and a Google Colab Research Credit Grant, as well as the support of the Rhodes Trust to Melas-Kyriazi.

References

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Eleftheria Briakou, Sweta Agrawal, Joel Tetreault, and Marine Carpuat. 2021. Evaluating the Evaluation Metrics for Style Transfer: A Case Study in Multilingual Formality Transfer. *arXiv preprint arXiv:2110.10668*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuan-Jing Huang. 2019. Style Transformer: Unpaired Text Style Transfer without Disentangled Latent Representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style Transfer in Text: Exploration and Evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830.

Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward Controlled Generation of Text. In *International conference on machine learning*, pages 1587–1596. PMLR.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2021. Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics*, pages 1–51.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating Unsupervised Style Transfer as Paraphrase Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762, Online. Association for Computational Linguistics.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-Attribute Text Rewriting. In *International Conference on Learning Representations*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, Retrieve, Generate: a Simple Approach to Sentiment and 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)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Fuli Luo, Peng Li, Pengcheng Yang, Jie Zhou, Yutong Tan, Baobao Chang, Zhifang Sui, and Xu Sun. 2019a. Towards Fine-Grained Text Sentiment Transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2013–2022.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019b. A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer. *arXiv preprint arXiv:1905.10060*.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabás Póczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020. Politeness Transfer: A Tag and Generate Approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881.

Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised Text Style Transfer with Padded Masked Language Models. *arXiv preprint arXiv:2010.01054*.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating Style Transfer for Text. *arXiv preprint arXiv:1904.02295*.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground Truth for Grammatical Error Correction Metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Shrimai Prabhumoye, Rafal Kocielnik, Mohammad Shoeybi, Anima Anandkumar, and Bryan Catanzaro. 2021. Few-shot Instruction Prompts for Pretrained Language Models to Detect Social Biases. *arXiv preprint arXiv:2112.07868*.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style Transfer Through Back-Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9.

Sudha Rao and Joel Tetreault. 2018. 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)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Machel Reid and Victor Zhong. 2021. LEWIS: Levenshtein Editing for Unsupervised Text Style Transfer. *arXiv preprint arXiv:2105.08206*.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A Recipe for Arbitrary Text Style Transfer with Large Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 837–848, Dublin, Ireland. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style Transfer from Non-Parallel Text by Cross-Alignment. *Advances in neural information processing systems*, 30.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "Transforming" Delete, Retrieve, Generate Approach for Controlled Text Style Transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3269–3279.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019. Harnessing Pre-Trained Neural Networks with Rules for Formality Style Transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3573–3578, Hong Kong, China. Association for Computational Linguistics.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2020. Formality Style Transfer with Shared Latent Space. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2236–2249.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. "Mask and Infill": Applying Masked Language Model to Sentiment Transfer. *arXiv preprint arXiv:1908.08039*.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired Sentiment-to-Sentiment Translation: A Cycled Reinforcement Learning Approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for Style. In *COLING*, pages 2899–2914.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Advances in neural information processing systems*, 32.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-Level Convolutional Networks for Text Classification. *Advances in neural information processing systems*, 28:649–657.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578.

A Appendix

A.1 Additional Details about Datasets

Previous TST studies have often chosen to focus on particular subtasks (such as changing the sentiment of a text from positive to negative) or particular datasets (such as YELP or AMAZON). In contrast, in our experiments, we decided to focus on a variety of TST datasets, some of which are known and widely used datasets in the field and some of which are new and synthetic. In the first half of this section, we present and discuss these datasets.[20]

Yelp Sentiment Dataset. YELP is a subset of the Yelp Review Polarity Dataset that was first used by Zhang et al. (2015) for a text classification task. It consists restaurant and other business reviews from Yelp, along with a label—either *positive* or *negative*—for each review. We used the version of the dataset that was curated by Li et al. (2018) in our experiments. The test set contains 500 positive and 500 negative samples, with one human reference (ground-truth) for each sample.

Amazon Sentiment Dataset. AMAZON is similar to YELP in its nature, but it contains product reviews that were obtained from Amazon. Each review is labeled either *positive* or *negative*. As before, we used the version of the dataset that was used by Li et al. (2018). The test set contains 500 positive and 500 negative sentences, with one human reference output for each sample.

Shakespearean English Dataset. We additionally used a small subset of the dataset that was used by Xu et al. (2012) originally for phrase-based machine translation, and experimented with "translating" sentences written in *Elizabethan* English to *modern* English. This small test set, which we call SHAKESPEARE, contains 599 paired sentences from William Shakespeare's *Romeo and Juliet*, written in Elizabethan and modern English.[21]

GYAFC Formality Dataset. Grammarly's Yahoo Answers Formality corpus (GYAFC; Rao and Tetreault (2018)) contains paired *informal* and *formal* sentences. Following Luo et al. (2019b), we used the samples from the "Family & Relationship" (F&R) domain and restricted our focus to the informal to formal direction. The test set contains 500 formal and 500 informal sentences.

JFLEG Corpus. The JHU FLuency-Extended GUG (JFLEG) Corpus was introduced by Napoles et al. (2017) to train and evaluate models for automatic grammatical error correction. It contains paired *grammatical* and *ungrammatical* sentences (with three error types—namely, awkward, orthographic, and grammatical). In our experiments, we focused on the ungrammatical to grammatical direction and used the publicly available test set that contains 747 sentences.

Symbolic Manipulation Task. We designed this small synthetic dataset to investigate how skillful the off-the-shelf language models are at writing symbolic expressions as natural English-language sentences. This dataset contains 1,000 example pairs, in which each input sample is written in a symbolic form (as either "$\alpha > \beta$" or "$\alpha < \beta$", where α and β are two different single words from the animal color, fruit, and number categories) and its corresponding output is basically the spoken utterance in English.

Remark. We realized that the original versions of all the aforementioned real-world TST datasets contain various tokenization issues (for instance, sentences sometimes contain extra whitespaces or have their puntuation marks separated out by spaces). We did not wish these tokenization artifacts to diminish the quality of our generations. To that end, we used a simple text-cleaning procedure to clean the texts before feeding them to our models.[22]

A.2 Additional Evaluation Metrics

Here, we describe in greater detail the standard automatic evaluation metrics used in the assessment of the performance of TST models.

Content Preservation. The standard metric for measuring semantic content preservation (or textual similarity, as we call it) has been BLEU (Papineni et al., 2002): If reference (ground-truth) sentences are available, then *reference*-BLEU scores are calculated by comparing model outputs to human-written ground-truth outputs using *n*-grams. Some recent studies (Lample et al., 2019; Dai et al.,

[20] We chose these datasets to broaden the semantic diversity of the TST tasks and to establish benchmarks for new TST studies. We share both the original and clean versions of some of the widely-used but poorly-tokenized datasets, such as AMAZON and YELP. In doing so, we hope to help address the recent call-to-action on reproducibility in TST from Jin et al. (2021); they encouraged researchers to share their data and evaluation codes in order to establish reliable benchmarks and facilitate easier comparison of new studies with existing work. We hope that our efforts will be a constructive step towards this goal.

[21] All the input sentences in SHAKESPEARE contain at least 10 and at most 25 words (inclusive).

[22] For the AMAZON and YELP datasets, we show the benefit of data-cleaning on overall performance. We also publicly release our text-cleaning code.

2019) further look at *self*-BLEU scores, comparing model outputs to input sentences—this is particularly done when reference sentences are not directly available. In our evaluations, we primarily used the SacreBLEU metric (Post, 2018)—as SacreBLEU has been shown to be a more reliable and accessible metric than BLEU—and considered both *reference*-SacreBLEU (*r*-sBLEU) and *self*-SacreBLEU (*s*-sBLEU) scores.[23] When evaluating the performances of models on the JFLEG corpus, we also used the sentence-level GLEU metric (Napoles et al., 2015), a variant of BLEU that was specifically designed for evaluating grammatical error correction (GEC) models.

Transfer Strength. To determine whether outputs generated by a TST model have the attributes of their target styles, the most common approach has been to train a (binary) classifier on the training set of the corpus of focus, where the sentences are taken as the inputs and their corresponding styles as the labels, and then to use this trained classifier to predict the percentage of the generated outputs for which the styles predicted by the model match their target styles.[24] In our sentiment transfer experiments, we measured transfer strength (sentiment accuracy) by fine-tuning pre-trained RoBERTa classifiers (Liu et al., 2019) on the training data in each case. In our experiments on SHAKESPEARE, we used the RoBERTa-based Shakespeare classifier of Krishna et al. (2020). Finally, in our experiments on GYAFC, we fine-tuned a pre-trained RoBERTa classifier on a subset of F&R examples.[25]

Fluency. With the emergence of successful LMs at our disposal, most recent TST models measure the fluency of their generated texts by computing perplexity (PPL) via a pre-trained LM like GPT-2.[26] Whilst this PPL-driven approach has the advantage of being automated and practical, it still contains considerable drawbacks, among which biases towards short texts and more frequent tokens can be listed right away. In our evaluations, we reported the average token-level PPL of generated texts using GPT-2-Large (774M).

[23] We used the SacreBLEU metric implemented in Hugging Face's *Metrics* library and lowered all the texts—both predictions and references—before calculating the scores.

[24] This method of measuring transfer accuracy demands access to either paired data for training a classifier or a pre-trained classier that can accurately estimate the style of an input text. It is, therefore, difficult to measure transfer accuracy for arbitrary or unknown styles, because there may not be any specific data to train a classifier.

[25] We release all our fine-tuned classifiers on our codebase.

[26] Early work used to measure fluency of sentences using an *n*-gram (typically trigram) Kneser-Ney language model.

A.3 Full Results

In the tables below, we include zero-shot results for the clean versions of AMAZON (Table 11) and YELP (Table 13), as well as the original versions of AMAZON (Table 10) and YELP (Table 12). We also include four-shot results for the clean versions of AMAZON (Table 14), YELP (Table 15), SHAKESPEARE (Table 16), JFLEG (Table 17), GYAFC (Table 18), and SYM (Table 19).

A.4 Further Discussion

Sentiment Transfer. Table 15 and Table 16 show the results for the clean versions of AMAZON and YELP, respectively. In terms of sentiment accuracy, GPT-2-XL yielded the best performance on both datasets, achieving 70% (87%) positive → negative accuracy and 56% (72%) negative → positive on AMAZON (YELP). In both cases, however, the sBLEU scores of GPT-2-XL were relatively lower than those of other models, indicating that it copied more from the source text. The GPT-Neo models had higher *r*-sBLEU and *s*-sBLEU scores than GPT-2-XL on both AMAZON and YELP, with only slightly worse accuracy scores. In the case of YELP-clean especially, the GPT-Neo/J models achieved good balances of sentiment accuracy, textual similarity, and fluency.

Shakespeare-to-Modern English Translation. As shown in Table 14, model performance generally improves with model size, with GPT-J-6B achieving almost 80% accuracy (according to the supervised classifier) and 21.9 *r*-sBLEU. Also notable is the difference between GPT-2-Small's high *s*-sBLEU score and low classifier accuracy, relative to the other models. Together, these indicate that the model copies large parts of the input text more often than the other GPT models.

Formality Transfer and Grammatical Error Correction. For GYAFC (Table 18), most models achieved accuracy scores above 80%, with increasing model size correlating with BLEU score. Notably, GPT-Neo-2.7B achieved an accuracy score of 81% and and a *r*-sBLEU score of 50 in the informal to formal direction. For JFLEG (Table 17), on the other hand, most models failed to outperform a simple baseline, which automatically copied the input text without making any changes. This baseline achieves a GLEU score of 37.2, better than all models except GPT-J (which obtains 40.0). Broadly,

there remains substantial room for improvement on JFLEG.

Symbolic Manipulation. Our final task is designed to measure the ability of these language models to copy and manipulate tokens under a refined synthetic experimental setup. With the exception of GPT-J, no model exceeded 60% accuracy on this synthetic dataset. GPT-J, by contrast, achieved 74% accuracy.

A.5 Additional Qualitative Examples

We provide additional qualitative examples from our language models in Tables 22-25.

A.6 Additional Related Work

Here, we describe additional related work on different subtasks of textual style transfer that could not be included in the main component of the paper due to space constraints.

These works can be broadly categorized into two families. The first family of approaches involves identifying and replacing distinctive style-related phrases (Li et al. (2018); Sudhakar et al. (2019); Wu et al. (2019); Madaan et al. (2020); Malmi et al. (2020); Reid and Zhong (2021), *inter alia*). For instance, Madaan et al. (2020) tackle the task of politeness transfer with a two-step text-editing approach, first identifying words with stylistic attributes using a n-gram TF-IDF method and then training a model to replace or augment these stylistic words with ones associated with the target attribute. Similarly, Li et al. (2018) propose a simple approach to sentiment and style transfer based on the idea that these attributes can often be identified by certain distinctive phrases. They identify these phrases, replace them with phrases associated with the target attribute, and combine them with an RNN to improve the fluency of the output text. Recently, Reid and Zhong (2021) propose to minimize the Levenshtein edit-distance between source and target texts, using a fine-tuned LM to make targeted edits. In general, these approaches perform well for very simple types of style transfer (e.g., negation by adding the word *not* to a sentence), but they struggle in scenarios that require more complex syntactic and semantic changes.

The second family of approaches involves disentangling latent representations of style and content Hu et al. (2017); Shen et al. (2017); Fu et al. (2018); Luo et al. (2019a); Wang et al. (2020) seek to learn a style-invariant representation for a piece of text, such that it can then be decoded in an arbitrary style. For example, Hu et al. (2017) encoded sentences into a style-agnostic space and then decode them in a style-specific manner using a variational autoencoder alongside attribute discriminators. Shen et al. (2017); Fu et al. (2018); Dai et al. (2019); Wang et al. (2019) improved upon this methodology through the use of cross-alignment, style embeddings, rule-based systems, and new architectures. While these approaches are often theoretically well-grounded, they generally require large quantities of labeled data and struggle with scaling beyond a small number of styles.

A.7 Computational Details

The computational cost of our experiments were quite low, as they only involve running inference on pre-trained models. All experiments were conducted on a single GPU. We usde an NVidia V100 for all experiments except those with GPT-J-6B, for which we used an RTX 8000 due to memory requirements. We estimate that all experiments for this paper consumed fewer than 30 GPU-days.

A.8 License Details

We will release all code for this experiment under an open-source license (MIT License).

A.9 Language Details

All datasets used for this paper are in English.

Dataset	[Few-Shot Examples] and [Test-Time Input]
AMAZON	Here is a text, which is positive: {very small but it works great in the car.} Here is a rewrite of the text, which is negative: {very small and it works terribly in the car.} \n ### \n Here is a text, which is positive: {i really loved it and will use it alot.} Here is a rewrite of the text, which is negative: {i really disliked it and will not use it again.} \n ### \n Here is a text, which is positive: {it gets the job done and for the price you can t beat it.} Here is a rewrite of the text, which is negative: {it does not work well and it was expensive.} \n ### \n Here is a text, which is negative: {i will never buy anything from this brand again.} Here is a rewrite of the text, which is positive: {i will buy from this brand again.} \n ### \n Here is a text, which is negative: {if your bike had a kickstand on the plate it won't lock down. } Here is a rewrite of the text, which is positive: {
YELP	Here is a text, which is negative: {this place is awful!} Here is a rewrite of the text, which is positive: {this place is amazing!} \n ### \n Here is a text, which is positive: {definitely will buy another pair of socks from this store–they have the best socks ever} Here is a rewrite of the text, which is negative: {definitely will NOT buy another pair of socks from this store–they have the worst socks ever} \n ### \n Here is a text, which is negative: {my wife and i were disappointed by the quality of the service–also, the food was pretty tasteless} Here is a rewrite of the text, which is positive: {my wife and i were impressive by the quality of the service–also, the food was pretty delicious} \n ### \n Here is a text, which is positive: {i loved their black tea and hot chocolate selections!} Here is a rewrite of the text, which is negative: {i hated their black tea and hot chocolate selections!} \n ### \n Here is a text, which is positive: {it's small yet they make you feel right at home.} Here is a rewrite of the text, which is negative: {
SHAKESPEARE	Here is a text, which is written in old English: {what hast thou there?} Here is a rewrite of the text, which is written in modern English: {what have you got there?} \n ### \n Here is a text, which is written in old English: {what say'st thou, my dear nurse?} Here is a rewrite of the text, which is written in modern English: {what did you say, my dear nurse?} \n ### \n Here is a text, which is written in old English: {and how doth she?} Here is a rewrite of the text, which is written in modern English: {and how is she doing?} \n ### \n Here is a text, which is written in old English: {talk not to me, for i'll not speak a word.} Here is a rewrite of the text, which is written in modern English: {don't talk to me, because i won't answer you.} \n ### \n Here is a text, which is old English: {as mine on hers, so hers is set on mine, and all combined, save what thou must combine by holy marriage.} Here is a rewrite of the text, which is modern English: {
GYAFC	\n Here is a text, which is informal: {sorry but donnt know if i can do this alone.} Here is a rewrite of the text, which is formal: {I am sorry, but I don't know if I can do this alone.} \n ### \n Here is a text, which is formal: {i am going to ask him to come to the concert with me, and i hope he accepts my invitation.} Here is a rewrite of the text, which is informal: {gonna ask him to come to the concert with me and hope he says yes :)} \n ### \n Here is a text, which is informal: {that sucks man but u gotta move on} Here is a rewrite of the text, which is formal: {that is unfortunate, but you need to move on} \n ### \n Here is a text, which is formal: {and i am sorry that you and your girlfriend broke up last week.} Here is a rewrite of the text, which is informal: {and im sorry that u and ur girlfriend broke up last week...} \n ### \n Here is a text, which is formal: {i mean that you have to really be her friend.} Here is a rewrite of the text, which is informal: {
JFLEG	\n Here is a text, which is ungrammatical: {There are several reason.} Here is a rewrite of the text, which is grammatical: {There are several reasons.} \n ### \n Here is a text, which is ungrammatical: {To my surprize nothing happened.} Here is a rewrite of the text, which is grammatical: {To my surprise, nothing happened.} \n ### \n Here is a text, which is ungrammatical: {This is important thing.} Here is a rewrite of the text, which is grammatical: {This is an important thing.} \n ### \n Here is a text, which is ungrammatical: {Water is needed for alive.} Here is a rewrite of the text, which is grammatical: {Water is necessary to live.} \n ### \n Here is a text, which is ungrammatical: {New and new technology has been introduced to the society.} Here is a rewrite of the text, which is grammatical: {
SYM	Here is a text, which is symbolic: {apple > seven} Here is a rewrite of the text, which is English: {apple is greater than seven} \n ### \n Here is a text, which is symbolic: {tiger < robin} Here is a rewrite of the text, which is English: {tiger is less than robin} \n ### \n Here is a text, which is symbolic: {teal > green} Here is a rewrite of the text, which is English: {teal is greater than green} \n ### \n Here is a text, which is symbolic: {apple < dog} Here is a rewrite of the text, which is English: {apple is less than dog} \n ### \n Here is a text, which is symbolic: {yellow > gray} Here is a rewrite of the text, which is English: {

Table 9: A complete list of example-prompts used in our few-shot experiments. Here, the color gray is used to highlight the examples used in our setups and the color teal an example test-time input in each specific TST task.

Model	Delimiter-Pair	Positive → Negative Acc	*r*-sBLEU	*s*-sBLEU	PPL	Negative → Positive Acc	*r*-sBLEU	*s*-sBLEU	PPL
GPT-2-Small (117M)	⟨·⟩	0.35	12.4	22.7	34	0.19	11.4	23.6	33
	* " · "	0.43	9.0	15.9	42	0.24	7.3	14.5	40
	⟩ " · "	0.46	6.6	11.3	29	0.23	6.8	14.1	30
	{·}	0.33	14.1	26.4	35	0.18	15.0	31.3	39
	– · –	0.40	6.8	12.6	29	0.17	6.5	13.8	26
	{{·}}	0.36	27.0	49.7	85	0.20	27.0	56.0	94
	(·)	0.35	18.1	32.7	54	0.18	17.6	38.2	59
	" · "	0.45	8.2	14.2	32	0.21	8.4	16.2	33
	[·]	0.35	18.9	35.5	60	0.21	14.3	29.3	43
	⟨⟨⟨·⟩⟩⟩	0.42	6.4	12.1	24	0.19	6.7	14.1	26
GPT-2-Medium (345M)	⟨·⟩	0.42	21.9	37.9	67	0.27	23.2	45.0	72
	* " · "	0.46	11.1	20.0	45	0.31	7.8	15.2	32
	⟩ " · "	0.45	13.4	22.4	43	0.29	6.2	13.4	27
	{·}	0.44	21.6	38.2	73	0.26	19.1	37.1	67
	– · –	0.63	4.2	7.0	22	0.31	3.7	7.6	21
	{{·}}	0.45	25.3	43.2	69	0.27	20.2	39.8	67
	(·)	0.49	19.4	32.4	69	0.31	18.1	35.5	69
	" · "	0.47	11.3	19.2	35	0.28	9.2	17.5	34
	[·]	0.54	17.1	28.6	63	0.32	13.1	26.3	52
	⟨⟨⟨·⟩⟩⟩	0.47	14.2	25.3	43	0.28	10.8	21.1	35
GPT-2-Large (774M)	⟨·⟩	0.38	25.5	43.7	52	0.24	27.8	58.0	73
	* " · "	0.39	27.1	46.7	72	0.25	22.9	47.0	60
	⟩ " · "	0.39	27.1	46.9	66	0.23	26.5	53.7	70
	{·}	0.39	28.7	48.8	77	0.24	28.0	54.8	63
	– · –	0.50	8.8	15.4	22	0.22	6.5	13.3	18
	{{·}}	0.43	27.7	46.5	63	0.25	36.6	69.3	113
	(·)	0.41	22.2	38.9	48	0.26	22.9	45.1	59
	" · "	0.52	19.7	31.4	57	0.30	17.7	34.2	48
	[·]	0.44	22.0	36.8	53	0.26	19.4	38.7	44
	⟨⟨⟨·⟩⟩⟩	0.47	13.1	21.9	28	0.28	12.2	24.6	25
GPT-2-XL (1558M)	⟨·⟩	0.40	26.3	43.0	81	0.31	25.9	48.2	82
	* " · "	0.42	22.7	39.3	60	0.29	17.6	33.3	44
	⟩ " · "	0.43	20.8	35.1	54	0.29	17.9	34.0	47
	{·}	0.47	23.8	37.6	73	0.31	22.7	42.5	80
	– · –	0.56	5.6	9.0	19	0.28	4.3	7.8	18
	{{·}}	0.42	29.8	49.8	99	0.29	25.7	46.4	80
	(·)	0.45	16.4	28.8	41	0.28	17.8	33.6	53
	" · "	0.47	16.1	26.2	38	0.30	14.6	28.4	41
	[·]	0.46	19.2	30.8	60	0.32	16.4	31.4	52
	⟨⟨⟨·⟩⟩⟩	0.51	9.0	13.9	25	0.37	7.4	12.7	26
GPT-Neo-1.3B (1.3B)	⟨·⟩	0.48	14.9	26.1	48	0.29	11.0	21.5	40
	* " · "	0.41	15.1	26.8	36	0.25	13.8	28.5	44
	⟩ " · "	0.38	19.7	36.0	60	0.26	18.9	37.4	48
	{·}	0.48	11.0	18.7	32	0.30	8.8	16.6	31
	– · –	0.54	5.0	8.6	18	0.30	4.9	9.6	18
	{{·}}	0.49	15.3	25.0	47	0.31	13.6	24.6	42
	(·)	0.44	14.3	25.3	44	0.27	15.5	29.1	51
	" · "	0.39	19.5	33.4	50	0.25	15.8	31.4	37
	[·]	0.46	15.1	27.0	51.7	0.26	15.1	30.4	47
	⟨⟨⟨·⟩⟩⟩	0.56	5.9	9.2	28	0.32	4.6	8.1	22
GPT-Neo-2.7B (2.7B)	⟨·⟩	0.43	20.0	36.1	53	0.27	22.2	43.8	59
	* " · "	0.37	26.2	46.7	65	0.21	26.8	54.1	65
	⟩ " · "	0.37	26.1	45.9	68	0.22	23.6	50.7	60
	{·}	0.44	21.6	37.7	61	0.29	27.1	51.4	80
	– · –	0.56	4.4	7.7	15	0.23	4.0	8.2	14
	{{·}}	0.42	23.7	42.0	56	0.24	29.5	58.5	72
	(·)	0.44	19.7	32.9	48	0.27	21.1	40.9	64
	" · "	0.38	26.1	44.5	67	0.22	26.3	52.7	67
	[·]	0.48	20.3	35.6	67	0.25	22.4	42.9	58
	⟨⟨⟨·⟩⟩⟩	0.45	14.1	24.8	32	0.22	21.0	42.1	55
GPT-J-6B (6B)	⟨·⟩	0.40	27.3	47.0	74	0.32	17.0	32.7	51
	* " · "	0.38	29.2	49.5	82	0.28	23.4	42.9	61
	⟩ " · "	0.36	27.4	47.2	69	0.30	23.1	43.6	64
	{·}	0.41	27.8	47.6	80	0.32	24.9	45.6	78
	– · –	0.43	7.1	12.3	19	0.20	4.8	9.0	17
	{{·}}	0.29	30.4	54.9	72	0.29	26.8	51.1	76
	(·)	0.48	24.9	41.6	80	0.35	22.3	39.3	77
	" · "	0.39	28.6	47.3	69	0.31	23.0	42.4	64
	[·]	0.43	23.3	38.2	63	0.37	20.8	38.3	60
	⟨⟨⟨·⟩⟩⟩	0.34	30.3	55.6	98	0.31	23.6	44.2	67

Table 10: Zero-shot performances of the off-the-shelf "small" language models from the GPT-2 and GPT-Neo/J families on the original version of the AMAZON dataset. Here, we also experimented with ten different delimiter-pairs, ranging from curly brackets to asterisk quotes: Overall, curly brackets $\{\cdot\}$, square brackets $[\cdot]$, parentheses $(\cdot)$, and quotes " · " yielded consistently reliable and high-quality outputs. Most of the models could not go beyond 60% accuracy in the positive to negative direction and 35% accuracy in the negative to positive direction. As shown in Table 11, most models performed marginally better (in terms of their accuracy, BLEU, and PPL scores) on the cleaner version of the dataset, suggesting that the original version might contain some tokenization-related (semantic) noises that might be preventing the models from performing well.

Model	Delimiter-Pair	Positive → Negative Acc	*r*-sBLEU	*s*-sBLEU	PPL	Negative → Positive Acc	*r*-sBLEU	*s*-sBLEU	PPL
GPT-2-Small (117M)	⟨·⟩	0.34	17.9	33.4	47	0.19	13.7	30.5	42
	* " · "	0.43	8.1	14.8	38	0.26	7.9	16.3	37
	⟩ " · "	0.45	6.3	11.6	29	0.25	7.7	15.4	32
	{·}	0.31	18.9	34.9	49	0.18	17.7	38.1	48
	– · –	0.42	6.6	12.0	24	0.21	6.6	13.7	26
	{{·}}	0.28	30.4	56.7	90	0.19	28.4	60.9	81
	(·)	0.34	22.2	39.1	57	0.21	18.7	40.1	48
	" · "	0.46	8.7	15.9	35	0.27	7.3	14.9	34
	[·]	0.32	19.9	35.4	59	0.20	18.4	39.7	55
	⟨⟨⟨·⟩⟩⟩	0.39	9.3	17.0	31	0.20	9.0	18.8	28
GPT-2-Medium (345M)	⟨·⟩	0.43	19.3	33.7	49	0.28	16.8	33.4	47
	* " · "	0.52	10.4	17.2	31	0.33	7.6	15.1	29
	⟩ " · "	0.46	10.4	17.6	30	0.32	6.3	12.9	25
	{·}	0.48	21.9	36.7	68	0.32	20.1	38.0	57
	– · –	0.57	3.9	6.8	20	0.29	3.1	5.9	18
	{{·}}	0.45	23.3	40.1	62	0.31	21.2	41.1	64
	(·)	0.45	19.9	33.1	50	0.32	17.6	35.3	58
	" · "	0.49	9.8	16.1	31	0.35	7.0	12.8	25
	[·]	0.50	15.7	25.9	42	0.30	13.2	26.1	46
	⟨⟨⟨·⟩⟩⟩	0.48	8.8	14.8	26	0.31	10.1	19.2	30
GPT-2-Large (774M)	⟨·⟩	0.40	26.3	44.6	68	0.29	22.9	43.4	52
	* " · "	0.44	23.5	40.9	54	0.27	18.8	36.9	42
	⟩ " · "	0.43	20.8	36.0	44	0.29	19.4	39.2	42
	{·}	0.44	28.6	49.1	70	0.28	26.0	51.2	55
	– · –	0.40	7.3	12.0	17	0.36	7.4	13.8	19
	{{·}}	0.43	31.0	53.1	90	0.29	31.3	60.0	79
	(·)	0.39	23.0	39.1	51	0.26	23.3	45.3	61
	" · "	0.47	18.1	29.8	47	0.31	16.8	32.7	48
	[·]	0.47	20.2	34.3	50	0.26	17.5	34.1	42
	⟨⟨⟨·⟩⟩⟩	0.49	9.9	16.2	21	0.27	9.4	18.2	22
GPT-2-XL (1558M)	⟨·⟩	0.40	25.6	42.0	68	0.29	23.1	43.0	65
	* " · "	0.36	22.5	39.4	48	0.31	18.7	37.5	47
	⟩ " · "	0.40	18.8	31.5	43	0.27	19.2	37.8	46
	{·}	0.46	21.5	35.4	59	0.32	22.3	41.4	70
	– · –	0.53	7.2	11.7	23	0.32	6.9	11.8	21
	{{·}}	0.45	25.7	43.2	81	0.31	24.8	45.4	72
	(·)	0.48	19.3	30.7	52	0.30	17.8	33.4	53
	" · "	0.45	20.5	33.6	49	0.31	17.9	33.4	51
	[·]	0.47	21.1	33.4	55	0.32	19.2	34.7	55
	⟨⟨⟨·⟩⟩⟩	0.47	7.8	13.1	24	0.38	6.8	12.5	25
GPT-Neo-1.3B (1.3B)	⟨·⟩	0.50	11.4	20.1	38	0.30	10.6	19.7	34
	* " · "	0.38	15.0	25.0	37	0.26	11.9	22.4	31
	⟩ " · "	0.40	12.6	22.1	35	0.26	11.3	22.2	29
	{·}	0.49	11.8	19.9	34	0.31	10.9	20.5	35
	– · –	0.50	4.1	6.8	18	0.25	4.4	8.5	18
	{{·}}	0.48	13.9	23.9	41	0.35	12.4	22.9	38
	(·)	0.42	16.6	27.8	53	0.28	13.1	25.8	42
	" · "	0.45	13.8	24.8	36	0.30	12.4	24.7	32
	[·]	0.46	16.7	28.1	45	0.26	14.7	28.5	43
	⟨⟨⟨·⟩⟩⟩	0.57	3.4	5.8	20	0.36	3.1	5.5	18
GPT-Neo-2.7B (2.7B)	⟨·⟩	0.44	20.1	34.8	51	0.29	19.5	38.3	46
	* " · "	0.40	27.2	47.9	61	0.22	28.9	57.6	58
	⟩ " · "	0.37	21.8	39.4	45	0.21	22.5	45.6	41
	{·}	0.48	21.4	36.5	55	0.28	23.7	45.9	57
	– · –	0.56	3.9	6.7	14	0.26	3.8	7.4	13
	{{·}}	0.43	21.2	36.2	44	0.27	28.0	55.7	56
	(·)	0.48	17.0	28.7	42	0.32	19.5	36.8	52
	" · "	0.38	25.6	44.5	58	0.22	28.6	58.2	59
	[·]	0.48	18.7	32.1	47	0.26	23.3	46.2	50
	⟨⟨⟨·⟩⟩⟩	0.49	14.8	24.9	33	0.32	18.7	37.3	37
GPT-J-6B (6B)	⟨·⟩	0.40	25.0	43.4	66	0.35	20.1	35.7	56
	* " · "	0.42	26.8	44.8	60	0.33	23.5	41.9	56
	⟩ " · "	0.39	29.3	50.3	65	0.31	24.2	44.8	64
	{·}	0.41	26.1	44.6	57	0.33	27.1	47.7	72
	– · –	0.46	5.9	9.7	17	0.23	4.6	8.6	17
	{{·}}	0.30	29.9	53.8	56	0.30	27.5	52.3	73
	(·)	0.44	21.0	34.8	53	0.37	19.5	37.7	64
	" · "	0.41	27.8	45.7	63	0.34	25.3	45.2	62
	[·]	0.45	20.1	34.0	53	0.36	20.8	37.7	65
	⟨⟨⟨·⟩⟩⟩	0.39	28.3	47.4	66	0.32	25.6	47.2	66

Table 11: Zero-shot performances of the off-the-shelf "small" language models on the clean version of the AMAZON dataset (AMAZON-clean, in short). As before, none of the models could go beyond the 60% accuracy level, but most of them seem to have achieved slightly better perplexity scores in the clean version of the dataset than in the original version.

Model	Delimiter-Pair	Positive → Negative Acc	*r*-sBLEU	*s*-sBLEU	PPL	Negative → Positive Acc	*r*-sBLEU	*s*-sBLEU	PPL
GPT-2-Small (117M)	⟨·⟩	0.38	10.1	29.7	49	0.11	13.8	40.6	47
	* " · "	0.41	3.1	10.6	31	0.21	4.4	14.0	33
	⟩ " · "	0.33	4.8	13.4	31	0.15	6.5	18.6	31
	{·}	0.36	8.4	23.3	35	0.15	7.8	23.5	34
	– · –	0.45	4.5	12.0	28	0.14	6.7	19.3	33
	{{·}}	0.37	13.4	38.9	62	0.11	16.6	49.1	54
	(·)	0.36	8.0	25.2	42	0.13	10.9	32.7	46
	" · "	0.37	4.9	13.7	30	0.18	6.7	20.3	35
	[·]	0.36	6.8	20.7	38	0.12	8.7	25.2	31
	⟨⟨⟨·⟩⟩⟩	0.43	2.1	5.8	17	0.09	2.1	6.4	17
GPT-2-Medium (345M)	⟨·⟩	0.55	9.9	27.0	44	0.31	10.3	28.2	38
	* " · "	0.64	7.4	17.5	40	0.38	7.2	16.7	33
	⟩ " · "	0.52	5.5	15.3	26	0.31	5.8	14.8	26
	{·}	0.66	7.6	17.6	34	0.35	8.7	24.1	34
	– · –	0.69	4.6	11.3	30	0.36	5.0	12.3	27
	{{·}}	0.68	12.2	30.3	52	0.32	14.8	36.6	58
	(·)	0.63	8.5	22.1	46	0.32	10.5	26.8	48
	" · "	0.66	5.1	13.1	29	0.41	6.3	15.2	30
	[·]	0.66	8.4	22.0	36	0.32	8.1	21.0	33
	⟨⟨⟨·⟩⟩⟩	0.64	1.8	4.7	16	0.24	2.2	5.9	16
GPT-2-Large (774M)	⟨·⟩	0.65	14.5	36.8	54	0.22	17.3	46.7	38
	* " · "	0.61	13.8	33.7	43	0.27	15.8	44.4	45
	⟩ " · "	0.57	16.2	44.0	59	0.27	18.7	51.4	53
	{·}	0.68	12.8	30.6	41	0.26	13.5	37.5	38
	– · –	0.64	8.9	22.3	31	0.24	9.7	26.3	25
	{{·}}	0.69	18.2	45.9	75	0.24	20.6	58.2	55
	(·)	0.68	10.6	26.0	40	0.28	15.4	40.6	46
	" · "	0.74	12.0	25.9	44	0.34	14.3	34.7	42
	[·]	0.70	8.3	20.2	31	0.28	9.7	26.2	32
	⟨⟨⟨·⟩⟩⟩	0.73	6.1	14.8	21	0.27	6.9	17.7	19
GPT-2-XL (1558M)	⟨·⟩	0.67	15.0	35.3	59	0.41	16.4	40.4	54
	* " · "	0.67	10.0	25.0	35	0.37	13.0	31.6	36
	⟩ " · "	0.66	10.4	25.8	41	0.34	12.5	30.2	39
	{·}	0.78	9.7	21.1	41	0.41	12.1	30.1	40
	– · –	0.74	6.4	13.9	25	0.37	6.3	14.2	20
	{{·}}	0.67	17.2	38.9	61	0.35	18.8	49.2	66
	(·)	0.72	8.6	18.5	35	0.40	12.4	28.3	42
	" · "	0.72	9.7	23.3	41	0.38	10.3	24.9	34
	[·]	0.72	9.2	22.0	35	0.41	10.1	23.5	31
	⟨⟨⟨·⟩⟩⟩	0.70	4.0	9.5	18	0.39	4.6	11.0	17
GPT-Neo-1.3B (1.3B)	⟨·⟩	0.61	6.5	16.0	28	0.38	6.8	15.7	26
	* " · "	0.31	13.3	38.7	33	0.24	13.5	35.4	37
	⟩ " · "	0.24	16.9	52.1	54	0.21	15.7	45.8	44
	{·}	0.66	3.2	8.4	19	0.38	5.3	12.2	21
	– · –	0.52	2.9	8.4	17	0.30	4.4	11.2	20
	{{·}}	0.60	9.1	23.6	35	0.39	8.5	21.2	30
	(·)	0.59	6.8	18.6	34	0.27	11.1	31.1	47
	" · "	0.46	14.9	40.2	54	0.23	14.9	40.1	47
	[·]	0.57	8.1	20.8	38	0.36	8.4	22.1	33
	⟨⟨⟨·⟩⟩⟩	0.68	1.3	3.8	17	0.38	1.9	4.1	16
GPT-Neo-2.7B (2.7B)	⟨·⟩	0.68	8.3	21.8	28	0.28	12.5	33.1	31
	* " · "	0.56	14.1	39.1	54	0.18	18.1	51.9	51
	⟩ " · "	0.54	12.1	34.3	43	0.21	15.9	46.4	42
	{·}	0.63	7.5	19.5	27	0.26	12.4	32.8	32
	– · –	0.63	3.4	8.5	16	0.26	3.7	9.8	14
	{{·}}	0.55	12.0	32.9	40	0.20	16.2	48.1	42
	(·)	0.73	7.3	18.0	38	0.34	13.6	34.2	47
	" · "	0.55	12.6	33.2	42	0.17	15.4	45.9	42
	[·]	0.62	8.0	21.3	33	0.27	13.7	37.0	39
	⟨⟨⟨·⟩⟩⟩	0.64	4.6	12.6	21	0.27	7.1	18.5	20
GPT-J-6B (6B)	⟨·⟩	0.60	14.9	37.6	52	0.44	14.4	33.0	51
	* " · "	0.57	16.2	41.0	57	0.36	16.1	37.1	49
	⟩ " · "	0.52	16.4	43.3	62	0.32	17.2	42.6	56
	{·}	0.60	13.6	36.1	51	0.46	14.3	32.0	46
	– · –	0.60	4.4	11.4	20	0.27	3.7	8.9	17
	{{·}}	0.44	16.2	44.4	50	0.34	17.8	43.0	56
	(·)	0.64	11.3	29.2	46	0.50	15.8	34.1	57
	" · "	0.57	12.7	34.0	52	0.37	14.8	34.5	43
	[·]	0.58	13.6	35.3	56	0.51	13.1	29.1	44
	⟨⟨⟨·⟩⟩⟩	0.47	16.1	45.3	58	0.40	13.5	31.7	43

Table 12: Zero-shot performances of the off-the-shelf "small" language models on the original version of the YELP dataset. Amongst all the model architectures, GPT-J-6B had the finest results, both quantitatively and qualitatively. We also note the performance differences between the positive to negative direction and the negative to positive direction across all the experiments. It appears that the former direction is easier for all the models than the latter direction. Furthermore, as in the case of AMAZON, Table 13 illustrates that most models performed slightly better in the the clean version of YELP than in the original version.

Model	Delimiter-Pair	Positive → Negative Acc	*r*-sBLEU	*s*-sBLEU	PPL	Negative → Positive Acc	*r*-sBLEU	*s*-sBLEU	PPL
GPT-2-Small (117M)	⟨·⟩	0.34	13.1	37.9	52	0.14	12.9	38.2	50
	* " · "	0.38	3.2	9.2	30	0.23	3.7	11.3	29
	⟩ " · "	0.38	3.1	10.4	25	0.16	5.3	14.9	27
	{·}	0.36	6.6	19.3	28	0.12	8.2	25.7	29
	– · –	0.43	4.3	12.4	24	0.12	5.3	16.7	27
	{{·}}	0.37	16.0	45.0	67	0.12	17.2	54.0	59
	(·)	0.42	8.2	22.5	34	0.13	12.1	37.9	37
	" · "	0.35	4.9	15.4	33	0.21	6.1	18.5	30
	[·]	0.42	8.3	23.0	41	0.13	11.3	35.9	39
	⟨⟨⟨·⟩⟩⟩	0.50	1.5	4.0	14	0.11	1.9	5.9	15
GPT-2-Medium (345M)	⟨·⟩	0.57	8.9	23.7	37	0.31	9.4	27.3	39
	* " · "	0.64	5.8	14.5	31	0.41	5.4	14.7	29
	⟩ " · "	0.52	5.7	16.7	28	0.30	5.6	15.0	26
	{·}	0.65	7.3	19.9	37	0.33	10.5	28.6	41
	– · –	0.66	3.9	9.7	23	0.34	3.0	7.4	20
	{{·}}	0.63	13.1	33.5	52	0.31	12.7	35.5	48
	(·)	0.64	9.4	25.3	44	0.29	11.8	33.8	44
	" · "	0.63	5.2	14.2	29	0.42	6.1	15.9	27
	[·]	0.64	7.0	18.4	35	0.33	8.2	22.4	33
	⟨⟨⟨·⟩⟩⟩	0.62	1.9	5.0	15	0.24	1.7	4.8	14
GPT-2-Large (774M)	⟨·⟩	0.63	14.3	36.3	46	0.27	17.9	48.2	44
	* " · "	0.65	13.5	33.3	47	0.35	12.5	34.6	36
	⟩ " · "	0.61	13.9	35.9	47	0.32	15.3	42.9	44
	{·}	0.67	12.0	28.8	40	0.30	12.5	33.8	30
	– · –	0.65	5.0	13.7	18	0.26	7.3	19.8	20
	{{·}}	0.75	17.2	39.9	59	0.31	21.3	58.1	62
	(·)	0.69	12.2	29.2	47	0.31	14.6	40.7	46
	" · "	0.77	11.8	27.3	41	0.37	11.7	29.6	34
	[·]	0.75	10.3	24.7	40	0.38	12.9	32.9	38
	⟨⟨⟨·⟩⟩⟩	0.72	3.6	9.1	16	0.31	4.2	10.7	15
GPT-2-XL (1558M)	⟨·⟩	0.64	17.4	40.1	58	0.35	17.3	41.5	53
	* " · "	0.69	11.3	28.2	40	0.41	12.6	31.3	33
	⟩ " · "	0.71	9.7	22.1	36	0.35	11.6	28.6	34
	{·}	0.73	8.6	21.3	35	0.46	11.4	25.9	35
	– · –	0.70	6.0	15.4	23	0.39	6.8	17.5	25
	{{·}}	0.63	17.4	40.9	70	0.38	19.1	46.8	59
	(·)	0.72	10.8	25.0	45	0.39	14.0	31.1	41
	" · "	0.77	7.6	17.6	31	0.44	9.9	23.1	30
	[·]	0.75	10.8	24.9	38	0.41	12.0	29.9	43
	⟨⟨⟨·⟩⟩⟩	0.68	2.2	5.4	14	0.32	2.0	5.1	13
GPT-Neo-1.3B (1.3B)	⟨·⟩	0.68	6.5	16.7	27	0.42	6.9	17.6	29
	* " · "	0.38	12.5	37.0	37	0.22	12.5	36.2	33
	⟩ " · "	0.32	13.7	42.3	41	0.19	16.1	47.9	40
	{·}	0.69	4.6	10.5	22	0.37	6.3	15.3	23
	– · –	0.58	3.1	8.1	18	0.33	4.2	11.1	17
	{{·}}	0.69	7.2	17.0	30	0.40	8.9	20.6	27
	(·)	0.63	8.6	21.3	39	0.28	8.3	23.3	28
	" · "	0.47	12.6	35.2	43	0.30	13.8	35.6	36
	[·]	0.68	8.3	21.4	40	0.34	8.6	23.9	30
	⟨⟨⟨·⟩⟩⟩	0.72	1.2	2.9	15	0.38	1.7	3.6	15
GPT-Neo-2.7B (2.7B)	⟨·⟩	0.66	8.8	23.4	31	0.32	13.9	35.3	36
	* " · "	0.58	14.5	36.9	42	0.17	17.4	51.1	42
	⟩ " · "	0.54	13.8	38.3	43	0.21	13.2	39.9	32
	{·}	0.64	7.0	19.0	24	0.28	11.0	32.6	29
	– · –	0.68	3.4	9.1	17	0.26	5.0	14.3	16
	{{·}}	0.57	11.0	29.3	31	0.24	15.5	46.1	35
	(·)	0.76	10.3	23.1	44	0.41	14.4	33.6	43
	" · "	0.59	12.5	33.6	42	0.21	14.9	43.4	38
	[·]	0.66	9.6	23.7	32	0.29	14.9	42.4	43
	⟨⟨⟨·⟩⟩⟩	0.64	5.4	14.4	21	0.27	8.1	21.7	23
GPT-J-6B (6B)	⟨·⟩	0.62	14.1	35.3	50	0.47	14.7	33.4	44
	* " · "	0.55	17.1	43.9	65	0.40	13.2	31.8	41
	⟩ " · "	0.61	16.9	41.6	56	0.38	13.3	30.5	37
	{·}	0.61	14.3	34.7	49	0.48	13.5	30.6	43
	– · –	0.54	5.4	14.7	22	0.36	4.8	10.9	19
	{{·}}	0.42	14.7	42.3	38	0.33	17.5	45.0	53
	(·)	0.66	12.9	30.7	50	0.51	11.5	23.4	44
	" · "	0.66	15.7	36.1	55	0.40	16.6	36.2	45
	[·]	0.69	11.8	28.7	45	0.53	13.3	27.3	43
	⟨⟨⟨·⟩⟩⟩	0.53	10.7	29.8	35	0.43	11.4	25.4	30

Table 13: Zero-shot performances of the off-the-shelf "small" language models on the clean version of the YELP dataset (YELP-clean, in short). In contrast to Table 12, we note that models, overall, achieved better results in YELP-clean than in YELP-original. Some models even could go beyond the 75% accuracy level in the positive to negative direction. Consistent with the previous findings, these results also indicate that curly brackets {·}, square brackets [·], parentheses (·), and quotes " · " are favourable delimiter-pairs, leading to better outcomes than many other delimiter-pairs.

Model	Setting	Shakespearean → Modern English Acc	*r*-sBLEU	*s*-sBLEU	PPL
GPT-2-Small (117M)	4-Shot (Top Choice)	0.35	17.1	42.4	65
GPT-2-Medium (345M)	4-Shot (Top Choice)	0.50	7.1	13.9	65
GPT-2-Large (774M)	4-Shot (Top Choice)	0.38	14.1	30.9	134
GPT-2-XL (1558M)	4-Shot (Top Choice)	0.39	18.9	38.4	90
GPT-Neo-1.3B (1.3B)	4-Shot (Top Choice)	0.39	17.2	37.0	63
GPT-Neo-2.7B (2.7B)	4-Shot (Top Choice)	0.62	23.9	41.4	106
GPT-J-6B (6B)	**4-Shot (Top Choice)**	**0.78**	**21.9**	**31.8**	**81**

Table 14: Four-shot performances of the off-the-shelf "small" language models on the clean version of the SHAKESPEARE corpus. In this few-shot setup, we included a simple natural-language task description and four illustrative examples in the prompt. We note that GPT-J-6B was able to "translate" sentences written in Elizabethan English to Modern English successfully, achieving a transfer accuracy score of 78%, reference BLEU score of 21.9, and perplexity value of 81.

Model	Setting	Positive → Negative Acc	*r*-sBLEU	*s*-sBLEU	PPL	Negative → Positive Acc	*r*-sBLEU	*s*-sBLEU	PPL
Style-Embedding	Li et al. (2018)	0.33	16.2	33.2	265	0.47	13.1	29.0	287
CrossAligned		0.66	2.2	3.0	93	0.74	1.7	2.4	96
DeleteAndRetrieve		0.49	33.3	60.3	120	0.51	26.7	53.5	113
TemplateBased		0.65	38.1	70.5	243	0.56	31.0	65.7	200
GPT-2-Small (117M)	0-Shot (Top Choice)	0.31	18.9	34.9	49	0.18	17.7	38.1	48
	4-Shot (Top Choice)	0.32	23.8	46.1	94	0.25	27.7	58.2	67
	4-Shot (RC: 3, IF)	**0.42**	**21.1**	**39.2**	**68**	**0.32**	**24.6**	**51.0**	**69**
	4-Shot (RC: 3, FS)	0.38	23.0	43.5	73	0.30	27.2	52.4	77
GPT-2-Medium (345M)	0-Shot (Top Choice)	0.48	21.9	36.7	68	0.32	20.1	38.0	57
	4-Shot (Top Choice)	0.44	12.3	17.8	78	0.42	11.5	17.8	72
	4-Shot (RC: 3, IF)	**0.58**	**12.6**	**18.6**	**66**	**0.55**	**10.2**	**15.0**	**53**
	4-Shot (RC: 3, FS)	0.52	7.2	10.0	59	0.50	5.6	9.0	56
GPT-2-Large (774M)	0-Shot (Top Choice)	0.44	28.6	49.1	70	0.28	26.0	51.2	55
	4-Shot (Top Choice)	0.47	17.1	27.7	54	0.32	15.0	27.4	101
	4-Shot (RC: 3, IF)	**0.60**	**15.4**	**24.7**	**62**	**0.43**	**15.6**	**27.7**	**59**
	4-Shot (RC: 3, FS)	0.55	21.6	33.1	55	0.35	20.0	34.7	53
GPT-2-XL (1558M)	0-Shot (Top Choice)	0.46	21.5	35.4	59	0.32	22.3	41.4	70
	4-Shot (Top Choice)	0.63	13.7	20.3	65	0.44	14.5	22.3	60
	4-Shot (RC: 3, IF)	**0.70**	**11.5**	**17.2**	**77**	**0.56**	**13.2**	**19.9**	**50**
	4-Shot (RC: 3, FS)	0.66	11.5	16.6	54	0.50	14.8	19.7	58
GPT-Neo-1.3B (1.3B)	0-Shot (Top Choice)	0.49	11.8	19.9	34	0.31	10.9	20.5	35
	4-Shot (Top Choice)	0.53	22.1	35.8	68	0.34	22.0	39.6	67
	4-Shot (RC: 3, IF)	**0.60**	**21.2**	**33.4**	**66**	**0.39**	**21.6**	**36.2**	**65**
	4-Shot (RC: 3, FS)	0.56	22.2	33.1	67	0.32	20.9	32.6	63
GPT-Neo-2.7B (2.7B)	0-Shot (Top Choice)	0.48	21.4	36.5	55	0.28	23.7	45.9	57
	4-Shot (Top Choice)	0.52	22.3	33.7	74	0.35	22.3	39.5	74
	4-Shot (RC: 3, IF)	**0.60**	**21.7**	**32.3**	**69**	**0.42**	**20.6**	**34.9**	**66**
	4-Shot (RC: 3, FS)	0.55	21.2	30.2	65	0.40	19.2	29.7	60
GPT-J-6B (6B)	0-Shot (Top Choice)	0.41	26.1	44.6	57	0.33	27.1	47.7	72
	4-Shot (Top Choice)	0.59	20.5	31.9	69	0.46	18.1	28.8	60
	4-Shot (RC: 3, IF)	**0.65**	**21.5**	**31.4**	**70**	**0.52**	**19.3**	**29.3**	**58**
	4-Shot (RC: 3, FS)	0.64	17.0	24.7	61	0.50	18.6	25.7	59

Table 15: Four-shot results on AMAZON-clean. We show the average results under (i) the zero-shot setting (*0-Shot (Top Choice)*); (ii) the four-shot setting in which we chose the top beam search result (*4-Shot (Top Choice)*; (iii) the four-shot setting in which we generated three outputs from the model, re-scored and re-ranked them according to the textual similarity and style factors—ignoring the fluency aspect—(*4-Shot (RC: 3, IF)*); and (iv) the four-shot setting in which we generated three outputs from the model, re-scored and re-ranked them according to the textual similarity, style, and fluency factors (*4-Shot (RC: 3, FS)*. (Here, "RC" denotes to the re-ranking-and-choosing method, "IF" ignoring fluency, and "FS" full set (meaning that we consider all the textual similarity, transfer accuracy, and fluency criteria). First, we stress that proving few-shot examples in the input resulted in 10-15% improvements in the accuracy scores in most of our models (see 0-shot results vs. 4-shot results). Second, we highlight that some of our off-the-shelf models (e.g., GPT-2-XL and GPT-J-6B) performed on par with, and even succeeded the performances of, the specially-tailored models of Li et al. (2018) along certain metrics. (For instance, our off-the-shelf models achieve significantly lower perplexity rates than theirs.) Third, we note that the Prompt-and-Rerank method (described in §3.2) seems to boost the models' performances in almost all the cases. Fourth, we note that 4-Shot (RC: 3, IF) often performs noticeably better than 4-Shot (RC: 3, FS) across all the models, suggesting that we may not need to include the fluency factor in our re-scoring calculations after all.

Model	Setting	Positive → Negative Acc	*r*-sBLEU	*s*-sBLEU	PPL	Negative → Positive Acc	*r*-sBLEU	*s*-sBLEU	PPL
BackTranslation	(Prabhumoye et al., 2018)	0.90	2.0	2.7	120	0.99	1.9	2.6	64
UnpairedRL	(Xu et al., 2018)	0.42	16.1	46.0	408	0.56	17.5	45.3	362
CrossAlignment	(Shen et al., 2017)	0.72	7.3	19.3	244	0.74	8.3	19.3	190
Multidecoder	(Fu et al., 2018)	0.42	13.4	43.2	376	0.49	12.6	35.5	369
StyleEmbedding		0.08	19.7	71.3	154	0.10	18.9	62.7	197
Style-Embedding	Li et al. (2018)	0.08	19.7	71.3	154	0.10	18.9	62.7	197
Delete-Only		0.89	12.7	33.1	195	0.81	14.0	34.7	169
Retrieve-Only		1.00	1.1	2.1	93	0.98	1.8	2.8	86
CrossAligned		0.72	7.3	19.3	244	0.74	8.3	19.3	190
DeleteAndRetrieve		**0.90**	**14.5**	**36.8**	**279**	**0.89**	**14.8**	**35.9**	**100**
TemplateBased		0.84	21.2	55.2	289	0.83	20.9	55.7	190
DualR	(Luo et al., 2019b)	0.91	26.5	58.7	125	0.85	25.3	58.8	141
B-GST	(Sudhakar et al., 2019)	0.83	19.8	46.8	153	0.79	23.4	46.1	163
Multi-Class	(**StyleTransformer**,	0.94	26.3	61.0	177	0.77	26.5	65.0	173
Conditional	Dai et al. (2019))	**0.95**	**22.6**	**52.6**	**211**	**0.87**	**23.1**	**53.0**	**234**
GPT-2-Small (117M)	**0-Shot (Top Choice)**	**0.36**	**6.6**	**19.3**	**28**	**0.12**	**8.2**	**25.7**	**29**
	4-Shot (Top Choice)	0.08	24.8	75.2	94	0.10	23.1	74.1	81
	4-Shot (RC: 3, IF)	0.14	21.5	71.0	73	0.17	18.5	58.5	99
	4-Shot (RC: 3, FS)	0.06	26.7	84.6	72	0.09	27.9	85.4	68
GPT-2-Medium (345M)	0-Shot (Top Choice)	0.65	7.3	19.9	37	0.33	10.5	28.6	41
	4-Shot (Top Choice)	0.49	14.5	34.1	72	0.35	13.3	35.2	56
	4-Shot (RC: 3, IF)	**0.68**	**15.0**	**35.1**	**69**	**0.53**	**12.6**	**29.8**	**45**
	4-Shot (RC: 3, FS)	0.43	20.4	46.4	74	0.40	17.7	43.5	48
GPT-2-Large (774M)	0-Shot (Top Choice)	0.67	12.0	28.8	40	0.30	12.5	33.8	30
	4-Shot (Top Choice)	0.79	16.6	32.8	84	0.57	14.5	31.0	74
	4-Shot (RC: 3, IF)	**0.79**	**10.6**	**24.7**	**79**	**0.58**	**12.1**	**30.3**	**53**
	4-Shot (RC: 3, FS)	0.58	23.0	56.8	76	0.45	22.1	53.5	64
GPT-2-XL (1558M)	0-Shot (Top Choice)	0.73	8.6	21.3	35	0.46	11.4	25.9	35
	4-Shot (Top Choice)	0.63	13.7	20.3	65	0.44	14.5	22.3	60
	4-Shot (RC: 3, IF)	**0.87**	**14.8**	**28.7**	**65**	**0.72**	**12.0**	**25.3**	**55**
	4-Shot (RC: 3, FS)	0.77	21.1	38.7	85	0.62	18.0	35.2	70
GPT-Neo-1.3B (1.3B)	0-Shot (Top Choice)	0.69	4.6	10.5	22	0.37	6.3	15.3	23
	4-Shot (Top Choice)	0.78	14.8	30.2	58	0.45	14.5	32.0	56
	4-Shot (RC: 3, IF)	**0.85**	**14.6**	**30.1**	**59**	**0.61**	**13.1**	**28.3**	**42**
	4-Shot (RC: 3, FS)	0.77	22.5	46.1	87	0.49	23.5	46.1	72
GPT-Neo-2.7B (2.7B)	0-Shot (Top Choice)	0.64	7.0	19.0	24	0.28	11.0	32.6	29
	4-Shot (Top Choice)	0.83	22.8	42.7	89	0.42	21.8	47.1	89
	4-Shot (RC: 3, IF)	**0.88**	**23.5**	**45.8**	**96**	**0.52**	**22.0**	**48.0**	**69**
	4-Shot (RC: 3, FS)	0.80	24.5	44.5	87	0.48	23.9	48.4	68
GPT-J-6B (6B)	0-Shot (Top Choice)	0.61	14.3	34.7	49	0.48	13.5	30.6	43
	4-Shot (Top Choice)	0.81	25.3	50.5	107	0.52	21.7	48.7	82
	4-Shot (RC: 3, IF)	**0.87**	**23.0**	**47.7**	**80**	**0.65**	**20.2**	**44.6**	**58**
	4-Shot (RC: 3, FS)	0.79	25.9	51.5	78	0.55	26.3	50.0	67

Table 16: Four-shot results on YELP-clean. As before, we detail the average results under different zero- and few-shot settings: Overall, our few-shot results on YELP-clean are consistent with those on AMAZON-clean, as reported in Table 15. GPT-2-XL and GPT-J-6B models, amongst all the models, have achieved the most successful performances, leveling themselves almost with the custom-made (trained) state-of-the-art models. We present some of the generated examples from these models in Table 21.

Model	Setup	Ungrammatical → Grammatical			
		GLEU	*r*-sBLEU	*s*-sBLEU	PPL
GPT-2-Small (117M)	4-Shot (Top Choice)	35.9	74.8	91.5	76
GPT-2-Medium (345M)	4-Shot (Top Choice)	19.9	38.0	40.6	63
GPT-2-Large (774M)	4-Shot (Top Choice)	30.0	56.8	64.1	55
GPT-2-XL (1558M)	4-Shot (Top Choice)	24.8	46.2	47.0	57
GPT-Neo-1.3B (1.3B)	4-Shot (Top Choice)	26.6	48.4	49.4	54
GPT-Neo-2.7B (2.7B)	4-Shot (Top Choice)	34.5	57.4	54.1	40
GPT-J-6B (6B)	**4-Shot (Top Choice)**	**40.0**	**64.8**	**59.1**	**48**

Table 17: Four-shot performances of the off-the-shelf "small" language models on the clean version of the JFLEG corpus. In this task, as a baseline, we consider the model which directly copies its input—we call this model "copy-input" model; this model achieves a GLEU score score 37.7. All but GPT-J-6B fail to beat the performance of the baseline "copy-input" model. GPT-J-6B, on the other hand, achieves a GLEU score of 40.0. Small language models fail, in fact rather miserably, at this grammatical error correction task. There is therefore an open room for improvement. We hope that our results will encourage researchers to come up with more effective ways to utilize pre-trained language models to solve this challenging problem.

Model	Setup	Informal → Formal			
		Accuracy	*r*-sBLEU	*s*-sBLEU	PPL
GPT-2-Small (117M)	4-Shot (Top Choice)	0.85	6.1	8.7	41
GPT-2-Medium (345M)	4-Shot (Top Choice)	0.76	12.9	16.2	39
GPT-2-Large (774M)	4-Shot (Top Choice)	0.78	23.2	31.3	33
GPT-2-XL (1558M)	4-Shot (Top Choice)	0.82	32.7	41.9	58
GPT-Neo-1.3B (1.3B)	**4-Shot (Top Choice)**	**0.85**	**36.4**	**49.6**	**68**
GPT-Neo-2.7B (2.7B)	4-Shot (Top Choice)	0.81	50.0	61.2	64
GPT-J-6B (6B)	4-Shot (Top Choice)	0.69	47.9	52.3	49

Table 18: Four-shot results on GYAFC-clean. We highlight that most of the off-the-shelf "small" language models could obtain at least 80% accuracy in the informal to formal direction. Amongst all the models, GPT-2-XL, GPT-Neo-1.3B, and GPT-Neo-2.7B appeared to be most successful, achieving not only high accuracy scores but also high BLEU scores and relatively low perplexity rates.

Model	Correct-Class Accuracy	Opposite-Class Accuracy	*reference*-sBLEU
GPT-2-Small	0.42	0.46	51.9
GPT-2-Medium	0.46	0.46	60.3
GPT-2-Large	0.53	0.35	65.6
GPT-2-XL	0.56	0.38	68.5
GPT-Neo-1.3B	0.55	0.37	67.3
GPT-Neo-2.7B	0.57	0.38	69.6
GPT-J-6B	**0.74**	**0.21**	**81.9**

Table 19: Four-shot performances of the off-the-shelf "small" language models on the symbolic manipulation task (SYM) defined in §5.1. Correct-Class accuracy refers to the accuracy of the model under exact-string matching, whereas Opposite-Class accuracy refers to the fraction of the cases for which the model copied and placed the right input words in the output but verbalized the incorrect (opposite) inequality symbol, that is writing "less than" instead of "greater than" or vice versa in between the expressions (for instance, the ground-truth might be "olive is greater than cat", but the model might have generated "olive is less than cat.") It was surprising to discover that most models failed to go beyond 60% accuracy on this small dataset. GPT-J-6B, on the other hand, outperformed all the other models, achieving an accuracy score of 74% on this task. We also remark that of the cases for which the models failed to generate the correct output, they often were able to copy the appropriate words from the input but failed to write the correct inequality symbol at the end.

Model	Setting	Positive → Negative				Negative → Positive			
		Acc	*r*-sBLEU	*s*-sBLEU	PPL	Acc	*r*-sBLEU	*s*-sBLEU	PPL
GPT-2-Small (117M)	4-Shot, Vanilla	0.14	25.3	82.1	79	0.13	24.9	80.6	75
	4-Shot, Contrastive	0.14	21.5	71. 0	73	0.17	18.5	58.5	99
	4-Shot, Negation-v1	**0.05**	**25.2**	**83.7**	**72**	**0.07**	**23.1**	**75.3**	**68**
	4-Shot, Negation-v2	0.06	25.4	84.5	75	0.08	25.3	83.0	74
GPT-2-Medium (345M)	4-Shot, Vanilla	0.63	19.7	49.4	75	0.38	17.7	49.7	62
	4-Shot, Contrastive	**0.68**	**15.0**	**35.1**	**69**	**0.53**	**12.6**	**29.8**	**45**
	4-Shot, Negation-v1	0.34	15.7	41.6	63	0.22	15.4	42.3	55
	4-Shot, Negation-v2	0.36	14.1	39.2	63	0.30	12.6	36.2	51
GPT-2-Large (774M)	4-Shot, Vanilla	0.75	16.2	38.9	60	0.52	17.2	45.6	55
	4-Shot, Contrastive	**0.79**	**10.6**	**24.7**	**79**	**0.58**	**12.1**	**30.3**	**53**
	4-Shot, Negation-v1	0.41	10.8	30.1	79	0.27	13.4	36.7	59
	4-Shot, Negation-v2	0.14	16.9	52.1	57	0.22	12.6	36.3	59
GPT-2-XL (1558M)	4-Shot, Vanilla	0.86	15.6	32.0	59	0.70	13.8	29.9	58
	4-Shot, Contrastive	**0.87**	**14.8**	**28.7**	**65**	**0.72**	**12.0**	**25.3**	**55**
	4-Shot, Negation-v1	0.83	11.8	24.1	81	0.50	14.9	32.0	54
	4-Shot, Negation-v2	0.53	19.0	43.5	77	0.51	16.9	37.7	61
GPT-Neo-1.3B (1.3B)	4-Shot, Vanilla	0.80	17.2	38.5	80	0.52	14.5	35.6	50
	4-Shot, Contrastive	**0.85**	**14.6**	**30.1**	**59**	**0.61**	**13.1**	**28.3**	**42**
	4-Shot, Negation-v1	0.79	16.1	34.7	72	0.00	0.0	0.0	0
	4-Shot, Negation-v2	0.57	16.5	40.6	67	0.00	0.0	0.0	0
GPT-Neo-2.7B (2.7B)	4-Shot, Vanilla	0.86	24.7	51.2	104	0.43	24.3	54.9	74
	4-Shot, Contrastive	**0.88**	**23.5**	**45.8**	**96**	**0.52**	**22.0**	**48.0**	**69**
	4-Shot, Negation-v1	0.80	22.5	47.8	79	0.00	0.0	0.0	0
	4-Shot, Negation-v2	0.76	22.0	48.2	85	0.00	0.0	0.0	0
GPT-J-6B (6B)	**4-Shot, Vanilla**	**0.90**	**23.5**	**51.0**	**85**	**0.62**	**22.8**	**49.9**	**63**
	4-Shot, Contrastive	0.87	23.0	47.7	80	0.65	20.2	44.6	58
	4-Shot, Negation-v1	0.82	23.6	50.9	85	0.53	24.3	54.7	65
	4-Shot, Negation-v2	0.73	25.1	59.1	95	0.48	25.2	58.1	66

Table 20: Four-shot performances of the off-the-shelf "small" language models on the YELP-clean dataset under different prompting protocols. Rows are bolded according to best accuracy scores. Across all models, the vanilla and contrastive prompting protocols appear to yield the most favourable results, though the contrastive prompting protocol typically performs better than the vanilla prompting protocol along the accuracy dimension, but not necessarily along the *r*-sBLEU dimension. We, however, note that in the case of GPT-J-6B, vanilla prompting performs on par with contrastive prompting overall.

Model	Text
Input *(P→N)*	the staff is awesome and location is right in the heart of old town!
GPT-2-Small	the staff is awesome and location is right in the heart of old town!
GPT-2-Medium	the staff was great but the location was an even higher priority.
GPT-2-Large	the staff is awful and location is not in the center of old town!
GPT-2-XL	the staff was kind of awful, location was kind of a shithole
GPT-Neo-1.3B	the staff is awful and location is just a couple minutes away from the city center!
GPT-Neo-2.7B	the staff is terrible and location is just outside of old town!
GPT-J-6B	the staff is terrible and location is in the middle of nowhere!
Input *(P→N)*	he always looks gorgeous and is so happy when i pick him up!
GPT-2-Small	he always looks gorgeous and is so happy when i pick him up!
GPT-2-Medium	he always looks so sad and upset by the fact that i pick him up!
GPT-2-Large	he always looks ugly and is so sad when i pick him up!
GPT-2-XL	he never looks gorgeous and is so sad when i pick him up!
GPT-Neo-1.3B	he always looks awful and is so sad when i pick him up!
GPT-Neo-2.7B	he always looks horrible and is so unhappy when i pick him up!
GPT-J-6B	he always looks ugly and is so sad when i pick him up!
Input *(N→P)*	the food was low quality and thrown together quickly with little flavor.
GPT-2-Small	the food was low quality and thrown together quickly with little flavor.
GPT-2-Medium	the food was high quality and thrown together quickly with a great flavor.
GPT-2-Large	the food was excellent and was cooked to perfection by very skilled and professional server-chefs.
GPT-2-XL	the food was high quality and tasted really good.
GPT-Neo-1.3B	the food was high quality and tasted amazing!
GPT-Neo-2.7B	the food was fantastic and presented exceptionally with excellent quality and excellent flavors!
GPT-J-6B	the food was delicious and prepared with a lot of care!
Input *(N→P)*	we went there on a busy holiday weekend and the service was extremely slow.
GPT-2-Small	we went there on a busy holiday weekend and the service was extremely slow.
GPT-2-Medium	we went there on a busy holiday weekend and the service was exceptional!
GPT-2-Large	we did go there on a busy holiday weekend–it was fast!
GPT-2-XL	we went there on a quiet weekend and the service was very fast.
GPT-Neo-1.3B	we went there on a busy holiday weekend and the service was exceptional!
GPT-Neo-2.7B	we went there on a slow holiday weekend and the service was extremely fast
GPT-J-6B	we went there on a busy holiday weekend and the service was extremely fast.
Input *(P→N)*	i love the outside seating and the reverse happy hour is incredibly affordable.
GPT-2-Small	i love the outside seating and the reverse happy hour is unbelievably affordable.
GPT-2-Medium	i love everything about this place–the music is great, everyone is super nice, and they have amazing desserts that are affordable.
GPT-2-Large	i loved the outside seating and the reverse happy hour is insanely expensive.
GPT-2-XL	i hate the outside seating and the reverse happy hour.
GPT-Neo-1.3B	i hate the outside seating and the reverse happy hour is incredibly costly.
GPT-Neo-2.7B	i hate the outside seating and the reverse happy hour is outrageously expensive.
GPT-J-6B	i hate the outside seating and the reverse happy hour is incredibly expensive.
Input *(N→P)*	the service is mediocre and the restaurant is just average.
GPT-2-Small	i love their black tea and hot chocolate selections!
GPT-2-Medium	i liked their black tea and hot chocolate selections; i was impressed by the quality of the food and service
GPT-2-Large	the service is wonderful!
GPT-2-XL	the service is great, and the restaurant is great!
GPT-Neo-1.3B	the service and the restaurant are just average.
GPT-Neo-2.7B	the service and the restaurant are just wonderful!
GPT-J-6B	the service is remarkable and the restaurant is amazing.

Table 21: Qualitative examples of sentiment transfer on YELP-clean. We manually highlight in blue and red phrases that we find to be associated with the input and target styles, respectively.

Model	Text
Input *(N→P)*	it works for maybe a few minutes and wears off quickly.
GPT-2-Small	it does not wear or work well when driving.
GPT-2-Medium	i think that it may have something to do with this brand.
GPT-2-Large	it works well for a while, but it wears off very quickly.
GPT-2-XL	it only does a few minutes' worth of work and then gives up.
GPT-Neo-1.3B	it can be used for a few minutes, but wears off quickly.
GPT-Neo-2.7B	it lasts a lifetime and wears off after a few hours.
GPT-J-6B	it works for a few hours and wears well.
Input *(N→P)*	i am only giving this item num_num stars because of the absurd price.
GPT-2-Small	i am only giving this item num_num stars because of the absurd price.
GPT-2-Medium	i will buy some num_num stars and will use them to have an extra day off when i work.
GPT-2-Large	i am only giving this item stars because it got a good review.
GPT-2-XL	i will NOT purchase this item again.
GPT-Neo-1.3B	i really liked it and i will do it more often.
GPT-Neo-2.7B	i am giving this item num_num stars, because of the absurd price.
GPT-J-6B	i am only giving this item 5 stars because it works perfect and the price is reasonable.
Input *(N→P)*	would have been alot nicer if it simply threaded in with the correct alignment.
GPT-2-Small	would have been alot nicer if it simply threaded in with the correct alignment.
GPT-2-Medium	it works perfectly.
GPT-2-Large	would have been so much better if the plastic was a bit thicker.
GPT-2-XL	would have been great if it just threaded in properly.
GPT-Neo-1.3B	would have been alot nicer if it simply threaded in with the correct alignment.
GPT-Neo-2.7B	it is very nice but it does not fit in properly at all.
GPT-J-6B	it works just fine and there is zero frustration with the way the threading is implemented.
Input *(P→N)*	it fulfills my modest requirements, and exceeds my expectations in appearance.
GPT-2-Small	it isn't good enough. I would buy from this brand again.
GPT-2-Medium	it doesn't fit my needs and falls short in other areas.
GPT-2-Large	it did not fulfill my modest requirements, and did not exceed my expectations in appearance, and would not fit well in the car.
GPT-2-XL	it does not meet my expectations in appearance.
GPT-Neo-1.3B	it does not fit my expectations and it does not fulfill my modest requirements in appearance.
GPT-Neo-2.7B	it doesn t fulfill my modest requirements, and i found it to be unsatisfactory.
GPT-J-6B	it fails to meet my needs, and is disappointing to look at.
Input *(P→N)*	great headset, solid build quality and blocks out the wind.
GPT-2-Small	i hate it
GPT-2-Medium	lots of dust on here and can be worn.
GPT-2-Large	not worth buying and not worth the price. It's just too large.
GPT-2-XL	i hate it, it is terrible.
GPT-Neo-1.3B	dreadful headset, terrible quality and wind blocks out the audio quality.
GPT-Neo-2.7B	lousy headset, hard to unblock the wind.
GPT-J-6B	great headset, but it does not work great in the wind.
Input *(P→N)*	i dropped phone once and the case held up perfectly.
GPT-2-Small	i dropped phone once and the case held up perfectly.
GPT-2-Medium	i dropped phone as a result.
GPT-2-Large	i never dropped my phone once and the case held up fine.
GPT-2-XL	i dropped phone twice and the case was damaged.
GPT-Neo-1.3B	i dropped it and it was very broken.
GPT-Neo-2.7B	i dropped phone as many times as I dropped a brick.
GPT-J-6B	i dropped the phone and the case broke into pieces.

Table 22: Qualitative examples of sentiment transfer on AMAZON-clean. We manually highlight in blue and red phrases that we find to be associated with the input and target styles, respectively.

Model	Text
Input	follow ur heart... it's neva wrong
GPT-2-Small	follow my heart.
GPT-2-Large	follow your heart... its nova wrong...
GPT-2-Medium	follow your heart... its nova wrong
GPT-2-XL	follow ur heart and be the man you always wanted to be and never had.
GPT-J-6B	follow your heart... it is never wrong
GPT-Neo-1.3B	follow your heart... its neva wrong
GPT-Neo-2.7B	follow your heart... it's never wrong
Input	if u you act like dat all da time she might think u desperate.
GPT-2-Small	if u acts like dat all, she might think that u wanna be happy with something and u dont want dat to break up with you, you will regret it, and she will become jealous. it is good...
GPT-2-Medium	if u u are always like dat..
GPT-2-Large	if you act like dat all da time she might think you desperate...
GPT-2-XL	if u u act like dat all da time she might think u desperate :)
GPT-Neo-1.3B	if u you act like dat everytime she thinks u desperately searching.
GPT-Neo-2.7B	if u you act like dat all da time she might think u desperate...
GPT-J-6B	if you act like that all the time she might think you desperate
Input	dont do things tht can get you hurt.
GPT-2-Small	but im sorry that u can do it and u can't leave
GPT-2-Medium	do things which i'm sure u will like.
GPT-2-Large	don't do things that can get you hurt
GPT-2-XL	dont do things that can end up getting u hurt.
GPT-Neo-1.3B	do things that can get you hurt.
GPT-Neo-2.7B	do not do things that can get you hurt.
GPT-J-6B	dont do things that might hurt you

Table 23: Qualitative examples of sentiment transfer on GYAFC-clean. We manually highlight in blue and red phrases that we find to be associated with the input and target styles, respectively.

Model	Text
Input	Fo example, in this genaration people watch movies and listen to musics more than read books and novels.
GPT-2-Small	Fo example, in this genaration people watch movies and listen to musics more than read books and novels.
GPT-2-Medium	In this generation people watch movies more than read books and novels.
GPT-2-Large	Fo example, in this genaration people watch movies and listen to musics more than read books and novels.
GPT-2-XL	Fo example, in this example people watch movies and listen to music more than read books.
GPT-Neo-1.3B	Example, in this genaration people watch movies and listen to musics more than read books and novels.
GPT-Neo-2.7B	Examples, in this generation, people watch movies and listen to music, whereas, reading books and novels is limited.
GPT-J-6B	For example, in this generation, people are watching movies and listening to musics more than reading books and novels.
Input	they are the boney and vital part o the society.
GPT-2-Small	They are the most important part of the society.
GPT-2-Medium	they are the vital part of the kingdom.
GPT-2-Large	they are the vital and boney parts of the society.
GPT-2-XL	they are the vital part, the boney part, and the society.
GPT-Neo-1.3B	they are the vital part of society.
GPT-Neo-2.7B	they are the boney and vital part of society.
GPT-J-6B	they are the bone and vital part of the society.
Input	And I am completely disagree with what the auther want to say.
GPT-2-Small	And I am completely disagree with what the auther want to say.
GPT-2-Medium	And I am entirely disagree with what the auther want to say.
GPT-2-Large	And I am completely disagree with what the auther want to say
GPT-2-XL	And I think the auther want to say completely differently.
GPT-Neo-1.3B	And I disagree with what he say.
GPT-Neo-2.7B	And I absolutely disagree with what the author want to say.
GPT-J-6B	I completely disagree with what the author wants to say.

Table 24: Qualitative examples on JFLEG-clean. We manually highlight in blue and red phrases that we find to be associated with the input and target styles, respectively.

Model	Text
Input	romeo, doff thy name; and for that name, which is no part of thee, take all myself
GPT-2-Small	speak you, my dear nurse, to me...
GPT-2-Medium	Romeo, Romeo, Romeo.
GPT-2-Large	romo, doff thy name, take all myself, all i'll say is, you're a dirty old man.
GPT-2-XL	roome, doff thy name; and for that name, which is no part of thee, take all myself.
GPT-Neo-1.3B	romeo, get out of my sight; get out, i'll not have it.
GPT-Neo-2.7B	rommy, get rid of your name; and because your name, which is no part of you, needs to get rid of, take all of me.
GPT-J-6B	romeo, take off your name, and for that name, which is not part of you, take all of me.
Input	now art thou sociable, now art thou romeo; now art thou what thou art, by art as well as by nature.
GPT-2-Small	now art thou romeo, now art thou romeo; now art thee what thou art, by art as well as by nature.
GPT-2-Medium	what I have said in thy speech thou hast said enough of me, wherewith to destroy thy soul.
GPT-2-Large	now art thou sociable, now art thou romeo; now art thou what thou art, by art as well as by nature.
GPT-2-XL	now is thou sociable, now is thou Romeo; now is thou what thou art, by nature as well as by nature.
GPT-Neo-1.3B	and now art thou what thou art, by art as well as by nature.
GPT-Neo-2.7B	now are you sociable, now are you romeo; now are you what you are, by nature.
GPT-J-6B	now you are friendly, now you are Romeo; now you are who you are, by nature as well as by art.
Input	the sweetest honey is loathsome in his own deliciousness and in the taste confounds the appetite.
GPT-2-Small	the sweetest honey is loathsome in his own deliciousness and in the taste confounds the appetite.
GPT-2-Medium	sweetest honey, sweetest bread, sweetest wine, sweetest dainties, do them to the soul's delights.
GPT-2-Large	the sweetest honey is loathsome in his own deliciousness; in the taste confounds the appetite.
GPT-2-XL	the most sweet and clearest honey is loathsome in the taste and in the very smell confounds the appetite.
GPT-Neo-1.3B	the honey is sweeter than ever it was in the honeycomb, and is almost enough to make one lose their senses and die a corpse.
GPT-Neo-2.7B	the sweetest milk is foul and sickly in his own deliciousness and in the taste confounds the appetite.
GPT-J-6B	the sweetest honey is disgusting in its own deliciousness and in its taste it confounds the appetite.
Input	unless philosophy can make a juliet, displant a town, reverse a prince's doom, it helps not, it prevails not.
GPT-2-Small	what am I thinking of?*
GPT-2-Medium	if philosophy can make a juliet, supplant your King, reverse his doom, reverse a prince's doom, reverse a king's doom.
GPT-2-Large	unless philosophy can make a juliet, displant a town, reverse a prince's doom, it helps not, it prevails not.
GPT-2-XL	unless it can convert a whore into a knight, it does not make it jolly.
GPT-Neo-1.3B	unless philosophy can make a juliet, a town, reverse a prince's doom, it helps not, it prevails not.
GPT-Neo-2.7B	unless philosophy can make a Juliet, cause a town to explode, reverse a prince's downfall, it helps not, it prevails not.
GPT-J-6B	unless philosophy can make a Juliet, overturn a town, reverse a prince's doom, it doesn't help, it doesn't work.
Input	night's candles are burnt out, and jocund day stands tiptoe on the misty mountain tops.
GPT-2-Small	not here
GPT-2-Medium	and when night comes, the sun sets and stars all flicker in the distance.
GPT-2-Large	night's candles are burnt out, and jocund day stands in a field of snow.
GPT-2-XL	night's candles are burning out, and jocund day is standing still on the misty mountain tops.
GPT-Neo-1.3B	night's fires are out and the day stands up above the foggy mountains.
GPT-Neo-2.7B	night's candles are burnt out; day's gone to bed and evening's at hand; the sun is setting on the mountain tops.
GPT-J-6B	day is over, and night is approaching, and it stands on its tiptoes over the misted mountain tops.
Input	nor tears nor prayers shall purchase out abuses, therefore use none.
GPT-2-Small	nor tears nor prayers shall purchase out abuses, therefore use none
GPT-2-Medium	nor tears nor prayers shall purchase out abuses, therefore use none.
GPT-2-Large	use none, for abuse is in the eyes of a man.
GPT-2-XL	no tears nor prayers shall buy out abuses, therefore I will not use any.
GPT-Neo-1.3B	nor tears nor prayers shall have any power, therefore use none.
GPT-Neo-2.7B	nor prayers nor tears will do any good, therefore use none.
GPT-J-6B	and tears and prayers won't work for you, so don't use them.
Input	and here stands all your state: either be gone before the watch be set, or by the break of day disguised from hence.
GPT-2-Small	and here stands all your state: either be gone before the watch be set, or by the break of day disguised from hence.
GPT-2-Medium	and this day there is nothing but the city.
GPT-2-Large	but where is the watch?
GPT-2-XL	and here stands all your state: either be gone before the watch, or by the break of day disguised from hence.
GPT-Neo-1.3B	and here stands all your state: either be gone before the watch be set, or by the break of day disguised from hence.
GPT-Neo-2.7B	and this is all you get: either go before the watch is set, or before the break of day dressed like a thief from hence.
GPT-J-6B	and here is all your stuff: either leave now or you'll have to deal with us when it's morning.

Table 25: Qualitative examples on SHAKESPEARE-clean. We manually highlight in blue and red phrases that we find to be associated with the input and target styles, respectively. (Footnote *: Pray tell us, what are you thinking of right now?)