Text Style Transfer: A Review and Experimental Evaluation

ZHIQIANG HU, University of Electronic Science and Technology of China
ROY KA-WEI LEE, Singapore University of Technology and Design
CHARU C. AGGARWAL, IBM T. J. Watson Research Center
ASTON ZHANG, Amazon Web Services

The stylistic properties of text have intrigued computational linguistics researchers in recent years. Specifically, researchers have investigated the Text style transfer (TST) task, which aims to change the stylistic properties of the text while retaining its style-independent content. Over the last few years, many novel TST algorithms have been developed, while the industry has leveraged these algorithms to enable exciting TST applications. The field of TST research has burgeoned because of this symbiosis. This article aims to provide a comprehensive review of recent research efforts on text style transfer. More concretely, we create a taxonomy to organize the TST models, and provide a comprehensive summary of the state of the art. We review the existing evaluation methodologies for TST tasks, and conduct a large-scale reproducibility study where we experimentally benchmark 19 state-of-the-art TST algorithms on two publicly available datasets. Finally, we expand on current trends and provide new perspectives on the new and exciting developments in the TST field.

CCS Concepts: • **Computing methodologies** → **Natural language generation**.

Additional Key Words and Phrases: Text Style Transfer, Natural Language Processing, Text Mining

ACM Reference Format:
Zhiqiang Hu, Roy Ka-Wei Lee, Charu C. Aggarwal, and Aston Zhang. 2020. Text Style Transfer: A Review and Experimental Evaluation. 1, 1 (April 2020), 45 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The stylistic properties of text have intrigued linguistic researchers for a long time. Enkvist [23] opined that text style is a "concept that is as common as it is elusive" and suggested that *style* may be described as a linguistic variation while preserving the conceptual content of the text. To give a practical example, the formality of text will vary across settings for similar content; examples include a conversation with friends such as "let's hang out on Sunday afternoon!", or a professional email such as "We will arrange a meeting on Sunday afternoon."

In recent years, the studies on text style have attracted not only the attention of the linguist but also many computer science researchers. Specifically, computer science researchers are investigating the text style transfer (TST) task, which is an increasingly popular branch of natural language generation [28] that aims to change the stylistic properties of the

Authors' addresses: Zhiqiang Hu, zhiqianghu@std.uestc.edu.cn, University of Electronic Science and Technology of China, 4E 2nd Section, 1st Ring Rd, Jianshe Road, Chenghua, Chengdu, Sichuan, 611731; Roy Ka-Wei Lee, roy_lee@sutd.edu.sg, Singapore University of Technology and Design, 8 Somapah Road, 05-301, Singapore, 487372; Charu C. Aggarwal, charu@us.ibm.com, IBM T. J. Watson Research Center, 1101 Kitchawan Rd, Yorktown Heights, New York, 10598; Aston Zhang, astonz@amazon.com, Amazon Web Services, 410 Terry Ave N., Seattle, Washington, 98109.

Manuscript submitted to ACM

text while retaining its style-independent content. Earlier TST studies have mainly attempted to perform TST with parallel corpus [10, 48, 50, 73, 89, 111, 130, 140, 141]. The parallel corpus in TST task consists of parallel sentences with different styles but same semantics. For instance, Xu et al. [141] proposed one of the first works to apply a phrase-based machine translation (PBMT) model to perform TST. They generated a parallel corpus of 30K sentence pairs by scraping the modern translations of Shakespearean plays and training a PBMT system to translate from modern English to Shakespearean English. For example, the Shakespearean sentence "*he slew thy kinsman*" and the modern English sentence "*he killed your relative*" are parallel sentences of different styles but similar semantics in the collected corpus. However, parallel data are scarce in many real-world TST applications, such as dialogue generation of different styles. The scarcity of parallel data motivated a new breed of TST algorithms that attempt to transfer text style without parallel data [13, 17, 27, 31, 35, 41, 46, 52, 65, 67, 72, 74, 76, 78, 88, 96, 112, 117, 122, 127, 132, 137, 138, 144, 145, 147, 149, 151, 152].

This survey aims to review the literature on the advances in TST thoroughly and provide experimental comparisons of various algorithms. It gives a panorama through which readers can quickly understand and step into the field of TST. It is noteworthy that the literature in the field is rather disparate, and a unified comparison is critical to aid in understanding the strengths and weaknesses of various methods. This survey lays the foundations of future innovations in TST and taps into the richness of this research area. To summarize, the key contributions of this survey are three-fold: (i) We investigate, classify, and summarize recent advances in the field of TST. (ii) We present several evaluation methods and experimentally compare different TST algorithms. (iii) We discuss the challenges in this field and propose possible directions on how to address them in future works.

The organization of this paper is as follows. We start our discussion on the related research areas that inspire the commonly used TST techniques in Section 2. Next, we explore and demonstrate some of the commercial applications of TST in Section 4. Section 3 provides the preliminary information about the TST. In Section 5, we categorize and explain the existing TST algorithms. The methodologies for evaluating TST algorithms are presented in Section 6. In Section 7, we present experiments on publicly available datasets to benchmark the existing TST algorithms. In Section 9, we outline the open issues in TST research and offer possible future TST research directions. Finally, we conclude in Section 10.

2 RELATED RESEARCH AREAS

TST finds its roots in the field of natural language generation and it is a relatively new research area. Many of the earlier TST works are also heavily influenced by two related research areas: *neural machine translation* [3, 16, 119] and *neural style transfer*, i.e., transferring styles in images [29, 51]. We found that a substantial number of TST techniques were adapted from the common methods used in natural language generation, neural machine translation, and neural style transfer. In addition, some of the evaluation metrics used in TST are also "inherited" from the natural language generation and neural machine translation tasks. In this section, we will provide some background on natural language generation and briefly introduce the two related research areas. Specifically, we will and highlight some of the common techniques and evaluation metrics that are transferred or adapted for the TST task.

2.1 Natural Language Generation

Natural language generation involves a broad range of natural language processing tasks that aim to generate coherent and semantically meaningful text from input data or machine representations. Such tasks include machine translation [6, 16], dialogue generation [75, 110], text summarization [30, 39, 83, 121], paraphrase generation [26, 79], visual storytelling [43, 129], and text style transfer [72, 112, 117, 132, 137, 149]. Recently, academia and industry have actively advanced natural language generation research, especially with the wide-adoption of large pre-trained language models [7, 20].

Many natural language generation tasks share common characteristics and goals. For example, most of these tasks aim to generate fluent texts with minimal grammatical error, and the generated texts should contain specific intended content. TST inherits these common goals and adds another objective: generating text in specific styles. Hence, TST models often extend natural language generation techniques to manipulate the style attribute in text. For example, generative adversarial network (GAN) has been applied to generate realistic and natural sentences [33, 42, 66]. Fu et al. [27] extended the GAN-based approach to disentangle texts' semantic and stylistic attributes to perform TST.

The evolution of TST methods closely follows the advancements in natural language generation techniques. For instance, [156] introduced a benchmarking platform for text generation models named *Texygen* which implemented a majority of text generation models and covered a set of metrics that evaluate the diversity, the quality, and the consistency of the generated texts. Nevertheless, TST also shared similar limitations as other natural language generation tasks. Specifically, there are shortcomings in existing evaluation methods used to assess the generated output by TST and natural language generation models [11, 84, 93]. For instance, there are currently no standard automatic evaluation metrics to assess the contextual quality or informativeness of generated text for specific natural language generation tasks. There is also little consensus on how human evaluations should be conducted.

2.2 Controllable Text Generation

Controllable text generation, which offers the possibility of controlling various aspects of generated textual content, has drawn much attention in the natural language generation research community in recent years. Aspects that are commonly controlled include context [109, 116, 126, 135], topic [22, 24, 128, 134], emotion [27, 64, 118, 154], user preferences [71, 77, 142, 143]. Naturally, the studies have also proposed controllable text generation techniques to control text stylistic proprieties and perform TST. Techniques, like GAN [32], reinforcement learning, VAE [59], and Transformer Vaswani et al. [123], which are employed in controllable text generation, could be also used in TST models. We will discuss some of these techniques in greater detail in Section 5.

2.3 Neural Machine Translation

Neural machine translation, a deep-learning-based approach for machine translation, is a well-studied research area [3, 16, 119]. Unlike the traditional statistical machine translation techniques [6, 63], neural machine translation can perform end-to-end training of a machine translation model without the need to deal with word alignments, translation rules, and complicated decoding algorithms. Both TST and neural machine translation are branches of natural language generation [28]. Naturally, the two research areas share a few similarities. First, two tasks are quite similar. Neural machine translation aims to change the language of a text sequence while preserving the content, and TST aims to modify the stylistic properties of a text sequence while also preserving the content. Second, most of the TST models have "borrowed" the most commonly used neural machine translation technique: the sequence-to-sequence encoder-decoder model [3, 16, 119]. Other TST studies have also adopted the back-translation technique originally proposed for neural machine translation [108] to transfer styles in texts [21, 98]. For instance, Prabhumoye et al. [98] used a back-translation model to extract the content features in text and subsequently generate text in different styles using the extracted content features and multiple decoders. Third, TST works have inherited some of the automatic quantitative evaluation metrics that were originally proposed to evaluate the neural machine translation task. For example, the bilingual evaluation understudy (BLEU) metric [95], which is used to evaluate the quality of machine-translated text by computing the modified n-gram precision between the generated and reference text, is also widely used in the TST task.

Despite the close similarities between TST and neural machine translation, it is also worth noting their subtle differences. The text attribute transferred in neural machine translation is the text language, which is definitive and easily observed. TST, on the other hand, focuses on transferring text style that is abstract and subtle. While both neural machine translation and TST are also concerned about preserving the original text's semantics and assuring the transferred text's fluency, most neural machine translation methods are style-independent. These differences motivated TST researchers to explore other techniques as such controllable generation [17, 41, 46, 67, 122, 147, 147] to ensure that the style in a text sequence is modified during TST. The need to evaluate if the style is effectively transferred also creates new evaluation metrics to assess the TST models [27, 84, 94].

2.4 Neural Style Transfer

Gatys et al. [29] first explored using a convolutional neural network (CNN) to extract content and style features from images separately. Their experimental results demonstrated that CNNs could extract content information from an arbitrary photograph and style information from a well-known artwork. Based on this finding, Gatys et al. [29] further experimented with the idea of exploiting CNN feature activation to recombine the content of a given photo with the style of famous artwork. Their proposed algorithm's underlying idea is to minimize the loss between the synthesized image's CNN latent representation and the desired CNN feature distributions, which is the combination of the photo's content feature representation and artwork's style feature representation. Interestingly, the algorithm also does not have any explicit restriction on the type of style images and does not require ground truth for training. The seminal work of Gatys et al. opened the new field of neural style transfer, which is the process of using neural networks to render content images in different styles [51].

The burgeoning research in the emerging field of neural style transfer has attracted wide attention from both academia and industry. In particular, natural language processing researchers are motivated to adopt similar strategies to implicitly disentangle the content and style features in text, and transfer the learned style features on another textual content [13, 27, 41, 52, 65, 65, 76, 96, 98, 112, 122, 144, 145, 145, 150–152]. For example, Fu et al. [27] proposed two TST models, which adopted an adversarial learning approach to implicitly disentangle the content and style in text. The first method used multiple decoders for each type of style to generate texts of different styles from a common content embedding. In the second approach, style embeddings are learned and augmented to a content embedding, and a single decoder is used to generate output in different styles.

While this line of TST works share similar objectives as neural style transfer approaches, disentangling content and style in texts has proven to be much harder than in the case of images [67]. First, the styles in images are easier to visualize and differentiate styles in two images in terms of patterns that can be modeled easily by a neural network. In contrast, text styles are somewhat more subtle, making it challenging to differentiate and define styles in two given pieces of text. Second, unlike the image's content and style, which are easily separated in the different CNN layers, the content and styles in the texts are tightly coupled and could not be not easily separated even with the style labels. Hence, some of the recent TST works have proposed a new direction to transfer text style transfer without disentanglement of text's content and style [17, 31, 35, 46, 67, 74, 78, 88, 127, 138, 147, 147].

3 STYLE AND NOTATION DEFINITION

Before we dive into the details of this survey, we first introduce the basic terminology and concepts used throughout this survey. We will also describe the TST task formulation summarize the notions commonly used in TST techniques.

3.1 Definition of Style

We discuss the definitions of text style from two perspectives: (a) the definitions of text style discussed in linguistics studies and (b) the definitions adopted in text style transfer literature.

Linguistic View. Style is an intuitive notion involving the manner in which some semantics is delivered [82]. The semantics (or '*content*' used in this survey) of text refers to the subject matter or the argument that the author wants to deliver. Text style is the literary element that describes the ways that how the author uses language, including word choice, sentence structure, figurative language, and sentence arrangement, all work together to establish tone, image and semantics in the text. Style indicates how the author describes events, objects, and ideas. Through style, the author can deliver additional information that the reader can interpret and respond to. It is impossible to list all possible styles. Every speaker has an idiosyncratic set of techniques, often tailored to particular hearers, for using language to achieve his or her interpersonal goals [38].

Style in Text Style Transfer. Unlike linguistics studies, which provide a more theoretical rule-based definition of text style, TST studies adopt a more data-driven approach in defining text style. Specifically, TST studies often consider '*style*' as the text style attributes or labels dependent on the style-specific corpora. These style-specific corpora usually contain text style attributes that are not difficult for neural machine learning techniques to model. For instance, sentiment transfer (i.e., *positive* and *negative* as style attributes) and formality transfer (i.e., *formal* and *informal* as style attributes) tasks are two of the most popular benchmarks for TST performance evaluation. We will provide a more comprehensive list of evaluation tasks and available corpus in Section 6. However, it is arguable that if the sentiment of a text can be regarded as its style. The main goal of TST is to modify the text's style while preserving its semantics. One could argue that sentiment transfer would have also modified the semantics of the sentence. Nevertheless, from the perspective of the current deep learning methods for TST, most of the existing TST models in this survey can be applied to any datasets labeled with different stylistics attributes, including sentiment, formality, gender, political slant.

3.2 Task Formulation

The TST task aims to change the stylistic properties of any given text while preserving its style-independent content. The input includes a set of attributes $\mathcal{A}$ with text for each attribute in the corpus $\boldsymbol{X}$. For example, for the formality transfer task, there are two attributes: formal and informal. The task is taking the sentence $\boldsymbol{x}$ with the source attribute s (e.g., formal) and generating the sentence $\boldsymbol{x'}$ with the target attribute t (e.g., informal) while preserving the style-independent content. The style corpus $\boldsymbol{X}$ can be parallel or non-parallel. In the parallel setting, for each sentence with the source attribute s, a counterpart sentence with the same style-independent content with the target attribute t is contained in $\boldsymbol{X}$. For the non-parallel setting, there is no alignment information among sentences with different attributes.

4 APPLICATIONS

The research on TST algorithms has many industrial applications and could lead to many commercial benefits. In this section, we summarize these applications and present some potential usages.

Table 1. Notation of each variable and its corresponding meaning.

Not.	Meaning
s	The source attribute value, e.g., the formal style
t	The target attribute value different from s, e.g., the informal style
$\mathcal{A}$	A predefined set of attribute values, $s, t \in \mathcal{A}$
$\boldsymbol{x}$	A sentence with the source attribute value s
$\boldsymbol{x}'$	The transferred sentence of $\boldsymbol{x}$ with the target attribute value t
$\boldsymbol{X}$	The corpus of sentences with different attribute values
E	Encoder of a TST model
G	Generator of a TST model
D	style classifier or discriminator
Θ_E	Parameters of the encoder
Θ_G	Parameters of the generator
Θ_D	Parameters of the style classifier
$\boldsymbol{z}$	Latent representation of text, i.e., $\boldsymbol{z} \triangleq E(\boldsymbol{x})$
$\boldsymbol{a}$	Latent representation of the attribute value in text

4.1 Writing Tools

One of the industrial applications of TST algorithms is the design of writing tools. Academics across various domains have widely researched Computer-aided writing tools, and the industry has developed many writing tool applications [60, 61, 81, 97, 114, 115]. The TST methods can be applied as new useful features in existing writing tool applications.

The utility of writing style has been widely studied by linguistic and literacy education scholars [2, 8, 34, 45, 54, 146]. The TST algorithms enable writing tool applications to apply the insights from existing linguistics studies to improve the writings of users. For instance, applying TST algorithms enables writing tool users to switch between writing styles for different audiences while preserving their writing content. The style evaluation methods developed to evaluate TST algorithms can also be applied to analyze the writing style of users [97]. For instance, the writing tool could analyze the style of users' business email draft to be too informal and recommend the users modify their writing to make the writing style more formal. Previous studies [9, 55, 90, 91, 106] have developed interesting real-world TST applications, where the texts are transferred between expert and layman styles. The underlying intuition is that expertise style transfer aims at improving the readability of a text by reducing the expertise level, such as explaining the complex terminology with simple phrases. On the other hand, it also aims to improve the expertise level based on context so that laymen's expressions can be more accurate and professional.

4.2 Persuasive Communication and Marketing

Studies have explored utilizing persuasive text to influence the attitude or behaviors of people [12, 56], and the insights gained from these studies have also been applied to improve marketing and advertising in the industry. The style of text has an impact on its persuasiveness [18, 53, 87], and the TST algorithms can be used to convert a text into a more persuasive style. Recent studies have also explored personalizing persuasive strategies according to the user's profile [57]. Similarly, TST algorithms could also be used to structure the text in different persuasive text styles that best appeal to the user profiles. For instance, TST algorithms can be applied to modify a marketing message into an authoritative style for users who appeal to authority. Jin et al. [49] proposed a compelling use-case to utilize TST methods to make news headlines more attractive. Specifically, a TST algorithm is used to transfer news headlines between humor, romance, and clickbaity style.

4.3 Chatbot Dialogue

The research and development of chatbots, i.e., intelligent dialogue systems that are able to engage in conversations with humans, has been one of the longest-running goals in artificial intelligence [1, 136, 155]. Kim et al. [58] conducted a study on the impact of chatbot's conversational style on users. They found that when informal conversational style was adopted, participants in the experiment were less likely to be persuaded to perform an action than those conversed with a formal-conversational-style chatbot. The study's encouraging results suggest that a chatbot's conversational styles may influence a user, and TST algorithms could be exploited to enhance the chatbots' flexibility in conversational styles. TST algorithms can be applied to equip chatbots with the ability to switch between conversational styles, making the chatbots more appealing and engaging to the users. For instance, a chatbot recommending products to customers may adopt a more persuasive conversational style while the same chatbot may switch to a formal conversational style when addressing the complaints from customers.

5 A TAXONOMY OF TEXT STYLE TRANSFER METHODS

In this section, we first propose a taxonomy to organize the most notable and promising advances in TST research in recent years. Then we discuss each category of TST models in greater detail.

5.1 Categories of Text Style Transfer Models

To provide an overview of this field, we classify the existing TST models based on types of (1) data settings, (2) strategies, and (3) techniques. Fig. 1 summarizes the taxonomy of TST methods. We consider TST as one of the many natural language generation tasks. The TST techniques developed can be broadly categorized by the data settings used in training, i.e., parallel supervised, non-parallel supervised, and purely unsupervised. As the bulk of the recent developments focuses on designing non-parallel supervised TST techniques, we scope our survey to cover these techniques in greater detail. Specifically, we discuss the broad strategies applied by the non-parallel supervised TST techniques and describe how the various techniques are designed to perform TST.

5.1.1 Data Settings. We broadly classify the existing TST studies into three categories based on the types of data settings used for model training.

- **Parallel Supervised.** In this data setting, the TST models are trained with known pairs of text with different styles. Commonly, NMT methods such as *sequence-to-sequence* (Seq2Seq) models [3, 16, 119, 141] are applied to transfer the style of text. For example, Jhamtani et al. [48] trained a Seq2Seq model with a pointer network on a parallel corpus and applied the model to translate modern English phrases to Shakespearean English. Details of the techniques applied on parallel datasets will be discussed in Section 5.2
- **Non-Parallel Supervised.** TST models in the non-parallel supervised setting aim to transfer the style of text without any knowledge of matching text pairs in different styles. Most of the existing TST studies fall into this category due to scarcity of parallel datasets in real-world TST applications.
- **Purely Unsupervised.** In both parallel and non-parallel supervised data settings, the style labels are available to enable supervised training of the TST models. A more challenging setting would be purely unsupervised

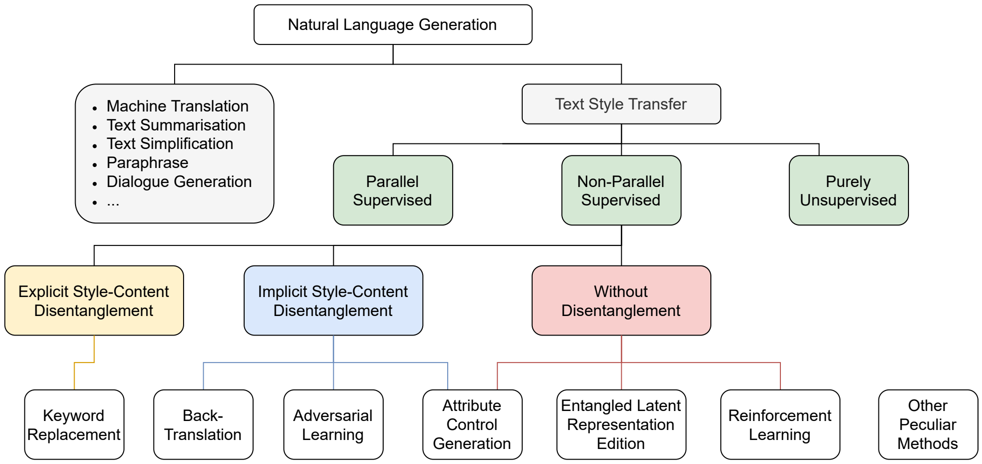

Fig. 1. A taxonomy of text style transfer methods

where only an unlabeled text corpus is available, and the TST models will be trained in an unsupervised fashion to perform text style transfer without any knowledge of style label.

5.1.2 Strategies. In order to perform TST in the *non-parallel supervised* setting, existing studies have proposed to disentangle the style and content in text, which is a strategy commonly used in NST [29].

The possibility of disentangling style and content has been widely discussed in the TST research community and remains an open research question. As discussed in Section 3, the semantics or content of text refers to the subject matter or the argument that the author wants to communicate, while text style describes the literary elements in which the text is communicated ways. The literary author includes word choice, syntactic structures, figurative language, and sentence arrangement. While the content and style have clear definitions, it is unclear if it is possible to disentangle text style and content. Intuitively, an author's stylistic choices could be influenced by the content communicated. For instance, considering the sentiment of texts about food reviews, some food may be universally considered tastier than others, e.g., most people love pizza. Thus, a food review that discusses pizza is likely to be associated with positive sentiment. With confounding factors like this, it becomes difficult to disentanglement style from the content without some additional assumptions or domain knowledge.

Lample et al. [67] also argued that it is challenging to perform style-content disentanglement and demonstrated that the adversarial method proposed by Fu et al. [27] did not separate style and content successfully. Nonetheless, Lample et al. [67]'s evaluation was only performed on the Fu et al. [27]'s model, and it is still an ongoing research question on whether we can disentangle style from content and how we can achieve this goal. We encourage researchers in TST community to explore the method to disentangle style and content more effectively and demonstrate what extent of the disentanglement has achieved from different perspectives.

In this survey, we discuss three types of style-content disentanglement strategies and categorizes the existing non-parallel supervised TST methods into one of these three strategies:

- **Explicit Style-Content Disentanglement.** In this strategy, the TST models adopt an explicit text replacement approach to generate texts of a target style. For instance, Li et al. [72] first explicitly identify parts of the text that is associated with the original style and then replace them with new phrases associated with the target style. The text with the replaced new phrases is then fed into a Seq2Seq model to generate a fluent text sequence in the target style. Details of the techniques applied to disentangle content and style will be explicitly discussed in Section 5.3.

- **Implicit Style-Content Disentanglement.** To disentangle style and content in text implicitly, TST models aim first to learn the latent representations of content and style of a given text sequence. Subsequently, the original text's content latent representation is combined with the latent representation of the target style to generate new text in the target style. Multiple techniques such as back-translation, adversarial learning, and controllable generation [27, 41, 98, 112, 151] have been proposed to disentangle the content and style latent representations.
- **Without Style-Content Disentanglement.** Recent studies have suggested that it is difficult to judge the quality of text style and content disentanglement, and the disentanglement is also unnecessary for TST [67]. Therefore, more recent research explored performing TST without disentangling the text's style and content. Techniques such as adversarial learning, controllable generation, reinforcement learning, probabilistic modeling, and pseudo-parallel corpus [17, 35, 67, 70, 78] have been applied to perform TST without disentanglement of the text's content and style.

5.1.3 Techniques. Table 2 lists various types of techniques that are commonly used to perform TST. We organize them following the previously mentioned taxonomy and will review each in detail in the following subsections. Additionally, we also present literature that adopted such techniques.

Table 2. Publications Based on Different Text Style Transfer Techniques

Data Setting	Strategy (Content-Style Disentanglement)	Technique	Literature
Parallel Supervised	-	Sequence-to-Sequence	[10, 48, 50, 73, 89, 111, 130, 140, 148]
Non-Parallel Supervised	Explicit	Explicit Style Keyword Replacement	[72, 117, 132, 137, 149]
	Implicit	Back-Translation	[98, 150]
		Adversarial Learning	[13, 27, 52, 65, 76, 96, 112, 144, 145, 151, 152]
		Attribute Control Generation	[41, 122]
		Other Peculiar Methods	[15]
	Without	Attribute Control Generation	[17, 46, 67, 147, 153]
		Entangled Latent Representation Edition	[74, 88, 127, 138]
		Reinforcement Learning	[31, 78]
		Other Peculiar Methods	[15, 35, 120]
Purely Unsupervised	-	Purely Unsupervised	[19, 100, 113, 138]

5.2 Sequence-to-Sequence Model with Parallel Data

The Sequence-to-Sequence (Seq2Seq) model [3, 16, 119, 141] based on the encoder-decoder architecture is core to many natural language generation tasks [28], where TST is no exception. Generally, a Seq2Seq model is trained on a parallel corpus, where the text of the original style is fed into the encoder, and the decoder outputs the corresponding text according to the target style. Various Seq2Seq TST models that are trained on parallel datasets have been proposed. Jhamtani et al. [48] extended the work of Xu et al. [141] by adding a pointer network [125] to the Seq2Seq model to selectively copy word tokens from the input text directly to transfer modern English to Shakespearean English. The direct copying mechanism is motivated by the intuition that Shakespearean English and modern English have significant vocabulary overlap, and there are infrequent nouns rare words that are harder to be generated by the Seq2Seq model. Carlson [10] proposed a Seq2Seq model with attention mechanisms and evaluated it with a collected parallel Bible-prose-style corpus.

However, applying Seq2Seq approach to the TST task is challenging because of the shortage of parallel data. To this end, researchers explored various data augmentation methods to enhance the parallel datasets used to train Seq2Seq TST models. For instance, many existing works have attempted to generate larger pseudo-parallel datasets to train Seq2Seq TST models. A commonly-used approach is to generate a pseudo-parallel dataset is to retrieve pseudo-references of a given text from a large text corpus [50, 73, 89, 111]. For example, Liao et al. [73] generated a pseudo-parallel dataset by searching pseudo-references containing similar content but different sentiments from Yelp reviews. For instance, given the sentence "*the food is terrible*", a matching pseudo-reference could be "*the food is delicious*", where "*the food*" is the similar content and the two sentences have very different review ratings, i.e., different sentiment. The generated pseudo-parallel dataset is subsequently used to train a Variational Autoencoder (VAE) [59] based framework to perform TST.

Besides retrieving pseudo-references from existing large text corpus to construct a pseudo-parallel dataset, Zhang et al. [148] had explored simply generating pseudo-references using a machine translation approach. Specifically, they first collected a set of potentially informal English sentences (e.g., from online forums) and subsequently translated these sentences into a pivot language (e.g., French), then translated them back into English. The back-translated English texts are further filtered using a pre-trained formality classifier, where the predicted formal text would be retained as the pseudo-reference for the informal texts.

The aforementioned data augmentation methods have demonstrated their effectiveness in improving parallel supervised TST models' training. Furthermore, the generated pseudo-parallel datasets could also be leveraged to trained and test other types of TST models. However, there remains a lack of systematic evaluation of the generated pseudo-parallel datasets; it is uncertain if the pseudo-references are indeed in a different style and retained the given text's content. Therefore, future TST pseudo-parallel data augmentation methods will need to consider the evaluation of the augmented data.

Another interesting data augmentation approach is to leverage and fine-tune large pre-trained language models to perform TST. For instance, Wang et al. [130] fine-tuned pre-trained GPT-2 model [102] using the text formality transfer rules harnessed from analyzing the GYFAC parallel dataset [103]. The fine-tuned GPT-2 model is subsequently used to transfer the formality of text (e.g., informal to formal text). This is a promising research direction as language models capture many facets of language relevant for many downstream natural language processing tasks, and they could be pre-trained with large text corpus. Future TST works may consider designing fine-tuning approaches that could better adapt the pre-trained language models for the TST task.

5.3 Explicit Style Keyword Replacement

Text style attributes such as sentiment are often marked by distinctive keywords and phrases. For example, words such as "nice" and "good" would infer positive sentiment, while words such as "bad" and "nasty" would infer negative sentiment. Therefore, an intuitive solution to transfer a text's sentiment would be to simply replace the distinctive keywords and phrases associated with specific sentiments. Motivated by this simple intuition, researchers have proposed

TST methods that explicitly disentangle content and style in text by replacing keywords attributed to specific style [72, 117, 132, 137, 149].

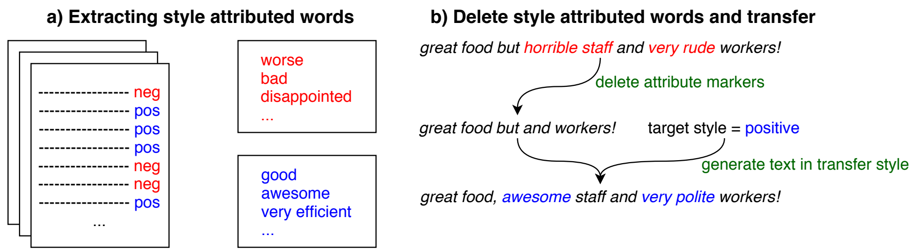

Fig. 2. An overview of the *Delete-Retreive-Generate* framework proposed by Li et al. [72]

.

A representative work of the explicit style keyword replacement approach is the *Delete-Retrieve-Generate* model [72]. Fig. 2 provides an overview of the *Delete-Retrieve-Generate* method. Given a text in source style (e.g., text with negative sentiment), the model first identifies the style-attributed words such as "*horrible, very rude*" in the text by computing the relative frequency of each word. The underlying intuition is that the style-attributed words are likely to be frequently used in sentences of specific style. Next, the model removes the style-attributed words from the text. The resulting text at this stage is assumed to contain only the content information such as "*food, staff*". Subsequently, the model retrieves a reference text similar to the source text with only content-related words. Note that the reference text would be retrieved from the target style corpus (i.e., text corpus with positive sentiment). The model then extracts the style-attributed words in the reference text using a similar approach. Finally, the retrieved reference text's style-attributed words are combined with the source text's content words to generate a text in target style using a rule-based approach or with a Seq2Seq model.

Existing explicit style keyword replacement TST methods mostly adopted similar framework, and focus on innovating the approach to identify and replace the style-attributed keywords. For example, Transformer [123] and deep learning text classifiers with attention mechanism [149] have been explored to identify and replace the style-attributed keywords [68, 117, 133]. These studies have shown that the advancement of the style-attributed keyword replacement methods would improve the TST task's performance. Recent works have also combined explicit style keyword replacement with cycled reinforcement learning to iteratively replace style attributed keywords while maintaining the content in text [132, 137].

The strength of explicit style keyword replacement TST approach lies in its simplicity; the models are relatively less complex with shorter training time. Furthermore, the keywords and phrases replacement also provides some explainability to the TST model; we can explicitly observe which part of the text is modified to transfer its style. However, this approach has several weaknesses. First, explicit style keyword replacement TST methods are mostly restricted to transferring the text's sentiment and cannot be applied to transfer other text style attributes such as formality. This is due to the nature of the sentiment transfer task, where certain lexicons often encode sentiment information, and replacing these lexicons is a viable approach to modify the text's sentiment. Transferring text's formality goes beyond simple keywords or phrases replacement because the text's formality could be encoded in its syntax. Second, it is challenging to apply explicit style keyword replacement TST to transfer beyond two text styles, i.e., positive to negative sentiment and vice versa.

5.4 Adversarial Learning

To overcome the limitations of explicit style keyword replacement methods, researchers have investigated another research direction to disentangle text's content and style information implicitly to perform TST. Adversarial learning is popular technique applied in many implicit content-style disentanglement TST methods [13, 27, 52, 65, 76, 96, 112, 144, 145, 151, 152]. Broadly, these TST methods apply adversarial learning for two purposes: (a) generate text output that are indistinguishable from the real data [13, 76, 96, 112, 122, 145, 152]; (b) removing the style attributes in text latent representation [27, 52, 65, 144, 151]. A representative work for later purpose is the seminal framework proposed by Fu et al. [27].

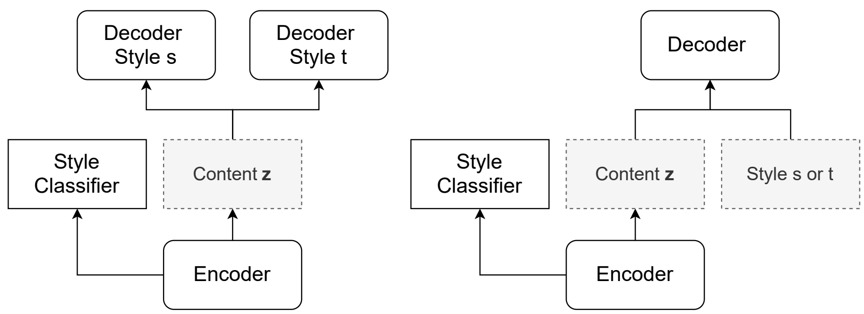

Fig. 3. Two common adversarial-learning-based TST models: multi-decoder (left) and style-embedding (right) proposed by Fu et al. [27]. The content representation c is the output of the of the encoder. The style classifier aims at distinguishing the style of the input. An adversarial network is used to ensure that content c does not have the style representation. In multi-decoder, multiple decoders are used to generate the text in specific styles. In style embedding, content representation c and style-embedding s are concatenated and fed into the decoder.

Fig. 3 illustrates two models included in the Fu et al. [27] proposed framework. In both models, an encoder is trained to generate intermediate latent representation of input text sequence $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T)$ of length T. An adversarial network is used to separate the content representation $\boldsymbol{z}$ from the style. The adversarial network is composed of two main components. The first component aims at classifying the style of input $\boldsymbol{x}$ given the representation learned by the encoder. The loss function to minimize is the negative log-likelihood of the style labels in the training data:

$$L_{\mathrm{adv1}}(\Theta_D) = -\sum_{i=1}^{M} \log p(l_i | \mathrm{Encoder}(\boldsymbol{x}_i; \Theta_E); \Theta_D),$$

where Θ_D and Θ_E are the parameters of the classifier and encoder, respectively. M denotes the size of the training dataset, and l_i refers to the style label. The second component aims at making the classifier unable to identify the style of input $\boldsymbol{x}$ by maximizing the entropy (i.e., minimizing the negative entropy) of the predicted style labels:

$$L_{\mathrm{adv2}}(\Theta_E) = -\sum_{i=1}^{M} \sum_{j=1}^{N} H(p(j | \mathrm{Encoder}(\boldsymbol{x}_i; \Theta_E); \Theta_D)),$$

where N is the number of styles. Note that the two parts of the adversarial network update different sets of parameters. They work together to make sure that the output of $\mathrm{Encoder}(\boldsymbol{x}_i; \Theta_E)$ does not contain style information.

Once the encoder is trained to produce the content representation, two generative approaches are used to generate the text in the target style. The first approach involves training multiple decoders (shown in Fig. 3-left) to take in the content representation and generate outputs in different styles. The second approach involves training a style embedding (shown in Fig. 3-right) and concatenating it to the content representation to output in target style using a decoder.

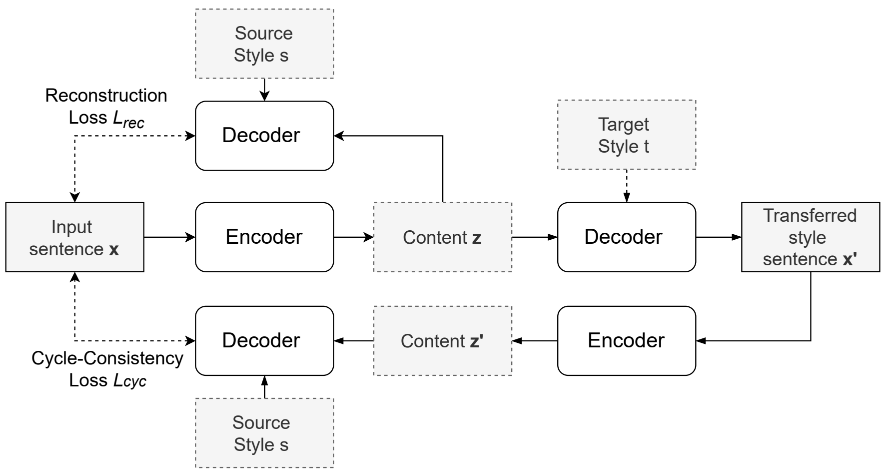

Fig. 4. Reconstruction loss L_{rec} and cycle-consistency loss L_{cyc} used in text style transfer.

While the proposed adversarial learning-based TST framework has demonstrated its effectiveness in disentangling the style from the content information in text implicitly, there remains uncertainty if the reconstruction loss [13, 27, 76, 112, 145, 151, 152] used its autoencoder architecture is sufficient to preserve the semantics of the input sentences. Researchers have proposed many variations of the adversarial learning-based TST framework to improve content presentation. For instance, studies have explored adding cycle-consistency loss [13, 52, 65, 76, 145, 152] to preserve the semantics of a style-transferred sentence. Fig. 4 illustrate the utilization of two the two losses. The reconstruction loss enforces the decoder, which takes the content representation z and original style embedding $\boldsymbol{a_o}$ as input, to reconstruct the input sentence x itself. This is a common approach applied in TST models utilizing an autoencoder architecture. To prevent model collapse during the TST operation, a cycle-consistency loss is used. Specifically, when a style-transferred sentence x' is fed into the TST model to transfer the sentence back to its original style, the cycle-consistency loss is used to enforce the generated sentences in the original style to be similar to the input sentence.

Besides employing loss functions, other adversarial learning-based TST methods have also proposed auxiliary components to enhance TST operation. Yin et al. [145] presented two partial comparators to guide adversarial learning, a content comparator that judges whether the input sentence and the generated sentence share the same content to improve content preservation, and a style comparator that judges if they have different styles. Lai et al. [65] combined the adversarial learning framework with a word-level conditional mechanism to preserve content information by retaining style-unrelated words while modifying the other style-related words. Yang et al. [144] replaced the style classifier in the adversarial learning framework with a target domain language model as the discriminator to provide richer and more stable token-level feedback during the adversarial learning process.

The adversarial learning-based models have demonstrated great performance in the TST task. However, it is not without criticism and weaknesses. Lample et al. [67] argued that the adversarial training is not effective on the disentanglement of style and content. They conducted experiments and demonstrated that a classifier trained on the learned content representation could recover the original style easily, suggesting the adversarial learning process may not have disentangled the style from the content representation. Another limitation of the adversarial learning-based TST technique is its dependency on the style classifier. The style classifier's accuracy limits the disentanglement of content and style; a poorly trained style classifier is unable to discriminate the style in a given text, thus affecting its disentanglement performance. Furthermore, large annotated datasets may be required to train the style classifier. Future adversarial learning-based TST models could explore solutions to address this limitation.

5.5 Back-Translation

Back-translation has been applied in NMT to generate artificial corpora [108]. Such an approach was also used to generate pseudo-parallel TST datasets [148]. However, this section reviews how back-translation has been explored to learn a content representation devoid of style for TST [98, 150]. This line of approaches is inspired by Rabinovich et al. [99], who found that the author's personal traits, such as gender, were obfuscated in human and machine translation. However, it is noteworthy that (1) Rabinovich et al. only showed that the gender information was obfuscated in human and machine translation, but text attributes like sentiment, tense, politeness were not explored; (2) no quantitative experiment was conducted to quantify the lost gender information.

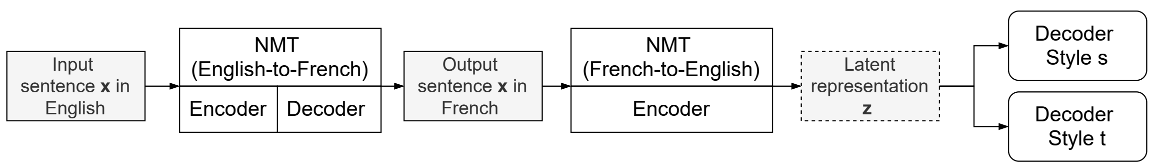

Fig. 5. Back-translation framework for TST proposed by Prabhumoye et al. [98]

Fig. 5 shows a back translation TST method adopted by Prabhumoye et al. [98]. The researchers attempted to use NMT models to rephrase the sentence and remove the stylistic information from the text. Specifically, an English text is first translated to French using an NMT model, then the translated French text is translated back to English using another NMT model. The latent representation z learned by the NMT model is assumed to contain only content information devoid of stylistic properties. Finally, the latent representation z is used to generate text in a different style using the multi-decoder approach. Zhang et al. [150] adopted an iterative back-translation pipeline to perform TST. The pipeline first learns a cross-domain word embedding in order to build an initial phrase-table, which is a set of corresponding phrases in the source and target styles. The phrase-table is then used to bootstrap an iterative back-translation model, which jointly trained two NMT systems to transfer text style.

Similar to the adversarial learning-based TST methods, the main concern with back-translation TST methods is its effectiveness in content-style disentanglement; it is uncertain if the learned content representation is indeed devoid of stylistic properties. It would be interesting to conduct a similar experiment proposed by Lample et al. [67] to investigate if it is possible to recover the style information from content representation. Another potential improvement to the back-translation TST methods is to improve the preservation of semantics in the learned content representation. A possible solution may be to add cycle-consistency loss to the existing back-translation TST methods [98, 150].

5.6 Attribute Control Generation

Attribute control generation is an increasingly popular technique used in TST models [17, 41, 46, 67, 122, 147, 153]. This technique often learn an attribute code $\boldsymbol{a}$ to control text generation in different styles. There are two broad strategies to applying attribute control generation to perform TST: (a) implicitly disentangling the latent representation $\boldsymbol{z}$ to contain only content information while learning the style attribute code; (b) without disentangling or constraining the latent representation $\boldsymbol{z}$ to contain only content information while learning the style attribute code. In both strategies, a classifier-guided loss is used to ensure that generator G generates a transferred sentence $\boldsymbol{x'}$ has the desired style attribute. Specifically, the loss function minimizes the following:

$$L_{\text{Cla}}(\Theta_G, t) = -\mathbb{E}_{p(\boldsymbol{x})}[\log D(\boldsymbol{x'})],$$

where D is a style classifier pre-trained on real data $\boldsymbol{x}$. Similar to the adversarial training on the generated sentences as introduced in Section 5.4, L_{Cla} can be trained with the the Gumbel-softmax distribution [47] or the policy gradient algorithm [131]. Note that the adversarial training on the generated sentences and the classifier-guided loss share the same goal: to ensure that the transferred sentence $\boldsymbol{x'}$ carries the target attribute t. However, the two approaches utilized different loss functions.

As the attribute control generation technique also depended heavily on the style classifier, this approach shares similar limitations as the adversarial learning-based method. Specifically, the style classifier's accuracy limits the effectiveness in learning a good style attribute code to perform TST. Nevertheless, if the style attribute code is effectively learned, it could be potentially transferred to enhance other NLG tasks. For example, pre-trained style attribute code can be leverage in machine translation methods to translate language and generate text in specific styles.

5.6.1 Attribute Control Generation with Content-Style Disentanglement. The pivotal study by Hu et al. [41] is a representative work for applying attribute control generation technique to implicitly perform content-style disentanglement for TST. Fig. 6 illustrates Hu et al. [41]'s attribute control generation TST model. The proposed model adopts a variational autoencoder (VAE) [59] framework. Unlike an autoencoder that learns a compressed representation for an input sequence, a VAE [59] learns the parameters of a probability distribution representing the data. The learned distribution can also be sampled to generate new data samples. Therefore, the generative nature of a VAE makes it widely explored and utilized in many natural language generation tasks [28]. In [41], the proposed a TST model utilized VAE to learn a sentence's latent representation $\boldsymbol{z}$ and leverage a style classifier to learn a style attribute vector $\boldsymbol{a}$. The probabilistic encoder of the VAE captures the variations of implicitly modeled aspects to guide the generator to avoid entanglement during attribute code manipulation. Finally, $\boldsymbol{z}$ and $\boldsymbol{a}$ are fed into a decoder to generate a sentence in the specific style. Specifically, the VAE loss function is shown as follows:

$$L_{\text{VAE}}(\Theta_G, \Theta_E; \boldsymbol{x}) = \text{KL}(q_E(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) - E_{q_E(\boldsymbol{z}|\boldsymbol{x})q_D(\boldsymbol{a}|\boldsymbol{x})}[\log p_G(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{a})],$$

where $\text{KL}(\cdot||\cdot)$ is the KL-divergence, and Θ_G and Θ_E denote the parameters of the decoder and encoder, respectively. The conditional probabilistic encoder E, denoted as $q_E(\boldsymbol{z}|\boldsymbol{x})$, infers the latent representation $\boldsymbol{z}$ given input sentence $\boldsymbol{x}$, and $q_D(s|\boldsymbol{x})$ is the conditional distribution defined by the classifier D for each structured variable in $\boldsymbol{a}$. To ensure that $\boldsymbol{z}$ retains only the style-independent content information, a *independency constraint* is proposed to ensure that the latent representation $\boldsymbol{z}$ of the input sentence $\boldsymbol{x}$ and transferred style sentence $\boldsymbol{x'}$ remain close to each other. The added *independency constraint* essentially ensures that the content information is disentangled from the input text and encode in the latent representation z.

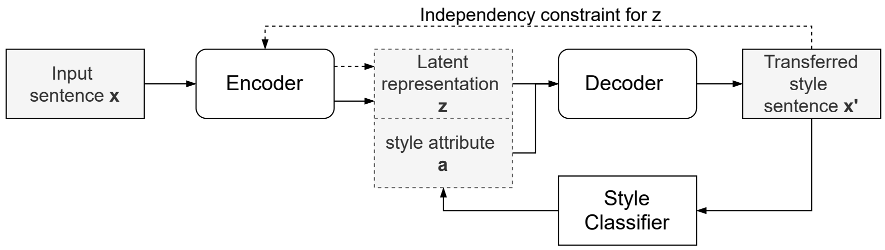

Fig. 6. Attribute controlled generation proposed in by Hu et al. [41]

It is worth noting that the training sequential VAE models has proven to be very challenging because of the posterior collapse problem [4, 14]. Annealing techniques are generally used to address this issue. Nevertheless, reconstructions from these models tend to differ from the input sequence. To address this limitation, Tian et al. [122] improve the content presentation of [41] by adding more constraints to preserve the style-independent content using part-of-speech (POS) information and a content conditional language model. Specifically, the researchers computed the distance between POS information of the input sentence and the output sentence as an error signal to the generator. This approach also assumes that the nouns of the sentence capture the content information. Therefore, they enforced the decoder to generate sentences with similar nouns.

5.6.2 Attribute Control Generation without Content-Style Disentanglement. As discussed in the earlier sections, Lample et al. [67] argued that it is challenging to disentangle content and style in text. Therefore, the researchers have proposed a new research direction to perform TST without content-style disentanglement. Fig. 7 illustrates the proposed model by Lample et al. [67]. The model employed denoising autoencoder [124] and back-translation [108] to build a translation between different styles. The noise function first corrupts the input sentence x by performing word drops and word order shuffling before feeding the corrupted output into an encoder to generate the latent representation z. Next, z is subsequently combined with a trainable target-style attribute a_t and input into a decoder to generate a sentence x' in target style. Finally, a back-translation process is initiated to have the generated sentence be fed into the sample encode-decoder process to reconstruct the original sentence using the latent representation z', and original style attribute a_s. The underlying intuition is that the decoder is forced to leverage on the target attribute a_s and a_t to generate the sentence x and x' as the latent representations z and z' only capture information of the corrupted input sentences. It is worth noting the proposed model did not contest or constrain that the latent representation z only captures style-independent content information. Therefore, Lample et al. [67] argued that this approach is able to perform TST without disentangling the content and style information in texts.

There are other variants of attribute-controlled approaches that perform TST without content and style disentanglement. For instance, Dai et al. [17] adopted a Transformer-based autoencoder [123] to perform TST with a trainable style attribute. The model's goal is to leverage the power of the attention mechanism in the Transformer to achieve better style transfer and better content preservation. Zhang et al. [147] proposed a shared-private encoder-decoder (SHAPED) framework that learns the style attributes to transfer the text style. Li et al. [70] extended the attribute-controlled TST

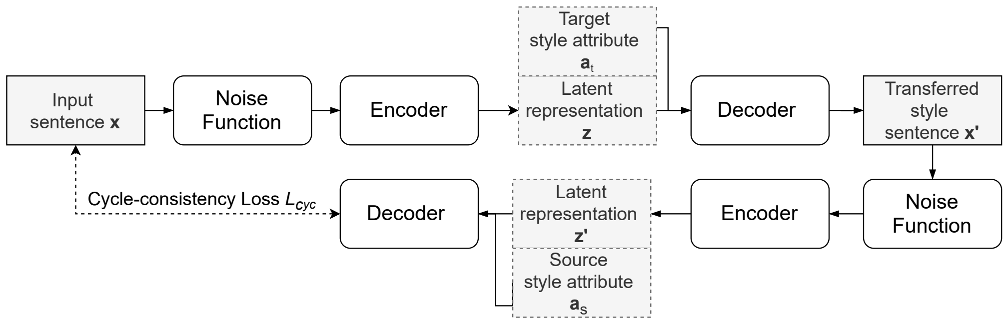

Fig. 7. Attribute controlled generation with back-translation proposed in by Lample et al. [67]

works and proposed a domain-adaptive TST that enables style transfer to be performed in a domain-aware manner. Specifically, besides the latent style attributes, the proposed model also learned domain vectors of the text in the source and target domain. The domain vectors are designed to represent different domains, such as movie reviews and restaurant reviews. This encourages the model to perform style transfer in a domain-aware manner instead of directly sharing the style transfer model. Intuitively, this design effectively avoids generations such as "The pizza is dramatic!" when domains are unspecified. The domain vectors are subsequently used with the style attributes and sentence's latent representations to perform TST across domains. Zhou et al. [153] proposed a fine-grained attribute-control method to perform TST. The proposed model utilized an attentional Seq2Seq model that dynamically exploits each output word's relevance to the target style for text style transfer. The model includes a carefully-designed objective function that fine-tuned the model's style transfer, style relevance consistency, content preservation, and fluency modeling loss terms. Dathathri et al. [19] proposed a language model for controllable language generation. The proposed method combines a pre-trained language model with one or more simple style classifiers that guide text generation without any further training of the language model.

5.7 Entangled Latent Representation Editing

Another line of work, which also attempted to perform TST without any content and style disentanglement, is directly editing the latent representation learned using the autoencoder-based models. Fig. 8 shows a common framework adopted by work that edits latent representations for TST. Typically, the latent representation z learned using an autoencoder is manipulated with various methods. Fig.8 shows a common framework used in editing text's latent representation for TST. A style classifier is jointly trained with the autoencoder, where the training iteratively updates the latent representation z in the constraint space to maximize the prediction confidence score for the target attribute label of this style classifier. Specifically, each update is calculated based on the gradient of style classifier loss with respect to z. The manipulated latent representation z' is then input into the decoder to generate text of the target style.

In earlier work, Mueller et al. [88] explored manipulating the hidden representation learned using VAEs to generate sentences that contain a certain style measured by a corresponding classifier. However, we noted that there was no quantitative evaluation of the effectiveness of text style transfer in this earlier work.

Xu et al. [138] conducted extensive experiments to investigate the latent vacancy in unsupervised learning of controllable representation when modeling text with VAEs. Similar to the study in [88], Xu et al. studied the impact on

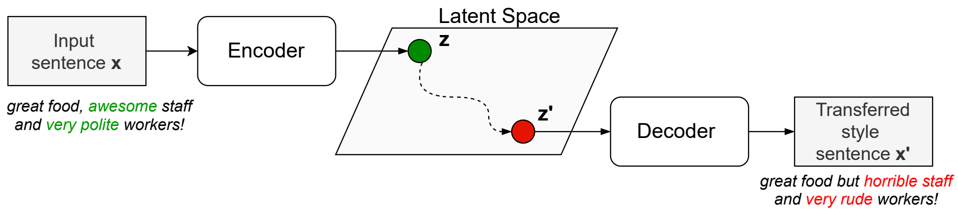

Fig. 8. Common framework for editing text's latent representation for TST.

text style when manipulating the factors in latent representations and both works found that if a manipulation fails at decoding accurate sentences, it is due to the manipulated results in representation areas that the decoder never saw during training. To handle this issue, both works proposed to perform latent-space manipulations within a constrained set. Specifically, [88] explicitly encourage the distribution of natural sequences is geometrically simple in the latent space by endowing $\boldsymbol{z}$ with their simple $N(0, \boldsymbol{I})$ prior. Whereas in [138] the researchers proposed to constrain the posterior mean to a learned probability simplex and only performed manipulation within the probability simplex.

Similar to the study in [88], Liu et al. [74] adopted a gradient-based optimization in the continuous space to manipulate the latent representation learned using VAEs and style classifiers to perform TST. Moreover, the proposed method naturally has the ability to simultaneously control multiple fine-grained attributes, such as sentence length and the presence of specific words, when performing TST tasks. Wang et al. [127] adopted a similar approach and performed a fast-gradient-iterative-modification algorithm to edit the latent representation learned using Transformer-based autoencoder until the generated text conforms to the target style.

The main challenge of latent representation editing TST methods is capturing the boundaries of the manipulations. As mentioned, both [138] and [88] observed that the decoder is not able to generate sentences in target-style if the manipulated results fall beyond the representation areas observed by the decoder during training. The current solutions focused on limiting the manipulation within a constrained latent space. However, it is unclear how the constrained latent space affects TST performance. Future research would need to explore better approaches to address this challenge.

5.8 Reinforcement Learning

Reinforcement learning has also been applied to perform TST. The core idea of reinforcement-learning-based TST is using the specially designed reward functions to guide the TST process instead of the various loss functions applied other TST methods. To optimize the parameters of the a reinforcement learning TST model, a policy gradient algorithm [131] is used to maximize the expected reward of the transferred text. The policy gradient algorithm makes the training easier without worrying about the difficulty of discrete training caused by the automatic regression decoding process. However, due to the high variance of the sampling gradient, training with this method may be unstable. For instance, Luo et al. [78] proposed to learn two Seq2Seq models between two styles via reinforcement learning, without disentangling style and content. Fig. 9 illustrates the proposed dual reinforcement learning framework. The authors considered the learning of source-to-target style and target-to-source style as a dual-task. The style classifier reward, R_s, and reconstruction reward, R_c, are designed to encourage style transfer accuracy and content preservation. The overall reward is the harmonic mean of the two rewards, and it is used as the feedback signal to guide learning in the dual-task structure. As such, the model can be trained via reinforcement learning without any use of parallel data or content-style disentanglement.

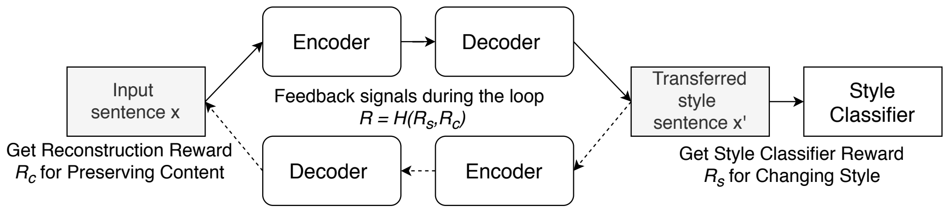

Fig. 9. Dual reinforcement learning model for TST proposed in [78].

Gong et al. [31] proposed a reinforcement-learning-based generator-evaluator framework to perform TST. Similar to previous TST works, the proposed model employs an attention-based encoder-decoder model to transfer and generate target style sentences. However, unlike the previous models that utilize a style classifier to guide the generation process, the proposed model employed a style classifier, semantic model, and a language model to provide style, semantic, and fluency rewards to guide the text generation, respectively. The key intuition is that the transfer of text style should not only ensure the transfer of style and content preservation but also generate fluent sentences.

Reinforcement learning is a promising research direction for TST. Although the existing reinforcement learning methods still heavily rely on style classifiers to get the TST process, the technique offers the possibility to design other reward functions to guide the transfer process. Computational linguistics researchers could design reward functions based on linguistics discourse theories to score text style and guide the TST process.

5.9 Purely Unsupervised Methods

Most of the TST studies discussed above are based on an assumption that style-specific corpora (i.e., parallel or non-parallel) are available. This section introduces another breed of TST models that performed style transfer in a purely unsupervised setting where only a mixed corpus of unspecified style is available.

There are relatively fewer works proposed to perform TST in purely unsupervised setting [46, 100, 113, 139]. In an earlier study, Radford et al. [100] explored the properties of byte-level recurrent language models where a LSTM model is trained on a text pre-processed as as a sequence of UTF-8 encoded bytes in an unsupervised fashion. More interestingly, the researchers discovered a single neuron unit within the trained LSTM model that directly corresponds to sentiment and manipulated neuron unit to transfer the sentiments in sentences. Xu et al. [139] conducted a similar study and successfully detect a latent dimension responsible for sentiment with 90+% accuracy. Subsequently, they use unsupervised representation learning to separate the style and content from a mixed corpus of unspecified styles and achieve satisfactory results on sentiment transfer task.

Jain et al. [46] propose an unsupervised training scheme to perform text formalization with unlabeled data. To train the encoder-decoder framework in an unsupervised manner, they employ external scorers to provide style information. These scorers, based on off-the-shelf language processing tools, decide the learning scheme of the encoder-decoder based on its actions. The authors label the formality level of sentences based on the scores given by the external scorers to help the TST model capture formality information.

Shen et al. [113] extended adversarial auto-encoders (AAE) with a denoising objective where original sentences are reconstructed from perturbed versions. The proposed denoising AAE model will map similar sentences to similar latent representations and make the boundaries of different representation clusters more obvious. To perform sentiment transfer, they compute a single "sentiment vector" by averaging the latent code z separately for 100 (non-parallel) positive sentences and negative sentences in the development set, and then calculating the difference between the two. Given a sentence from the test set, they attempt to change its sentiment from positive to negative or from negative to positive through simple addition/subtraction of the sentiment vector.

The competitive performances of the purely unsupervised TST methods are encouraging. However, we noted that most of the methods mentioned above were evaluated on the sentiment transfer task. Specifically, in [100] and [139], which have identified the latent attribute that corresponds to sentiment in text, it is unclear if the methods can be applied to other stylistic properties such as text formality. More studies would need to be conducted to evaluate if the purely unsupervised methods can be generalized to other TST tasks.

5.10 Other Peculiar Methods

TST is a relatively new research area that is constantly evolving, and there are peculiar methods that we are not able to sort into the techniques defined in our taxonomy framework. This section covers some of these TST methods.

He et al. [35] proposed a probabilistic deep generative model to infer the latent representations of sentences for TST. The proposed model hypothesizes a parallel latent sequence that generates each observed sequence, and the model learns to transform sequences from one domain to another in a non-parallel supervised setting. Specifically, the model combines a recurrent language model prior with an encoder-decoder transducer to infer the sentence's latent representations in the assumed parallel style corpus. The inferred latent representation is then used to generate the sentence of a specific style via a decoder. To the best of our knowledge, this is the first work that applied text transduction method on the TST task.

Unlike the other adversarial learning-based and attribute control generation methods that disentangled content and style information implicitly for TST, Cheng et al. [15] leveraged mutual information to achieve the same goal. Specifically, the researchers minimize the mutual information between style and content embedding to induce style and content embeddings into two independent low-dimensional spaces. Concurrently, they maximized the mutual information between content embedding and input sentence and the style embedding and style label to ensure that content embedding sufficiently captures content information from the input sentence and style embedding are strongly correlated to style label. Finally, to perform the TST task, two sentences are encoded into a disentangled representation, and the style embedding from one sentence and the content embedding from another are combined to generate a new sentence.

Syed et al. [120] proposed an interesting framework that adapts language models for non-parallel author-stylized rewriting. The researchers first pre-trained a language model on a large English corpus containing 141 authors' articles and Wikipedia articles in an unsupervised fashion using masked language modeling. Subsequently, they cascade the pre-trained language models with an encoder-decoder framework for text generation. The encoder-decoder is fine-tuned separately on each of the target author's corpus using Denoising Auto-Encoder loss:

$$L_{DAE} = \boldsymbol{E}_{\boldsymbol{x}\sim S}[-logP(\boldsymbol{x}|C(\boldsymbol{x}))] \quad (1)$$

where $C(\boldsymbol{x})$ is the noisy version which obtained by dropping and replacing works with special token $[BLANK]$ of the input sentence $\boldsymbol{x}$ and S are the sentence in target author's corpus. While performing TST of authors' writing style, the fine-tuned author-specific encoder-decoders are used to generate sentences in the target author's style.

Table 3. Dataset statistics for text style transfer.

Dataset	Subset	Attributes	#Text records
Shakespearean-Modern [48]	-	Shakespeare	21,076
		Modern	21,076
Yelp [112]	-	Positive	381,911
		Negative	252,343
IMDb [17]	-	Positive	181,869
		Negative	190,597
Amazon [36]	-	Positive	278,713
		Negative	279,284
GYAFC [103]	F&R	Informal	56,087
		Formal	55,233
	E&M	Informal	56,888
		Formal	56,033
PNTD [27]	-	Paper	107,538
		News	108,503
Caption [72]	-	Romantic	6,300
		Humorous	6,300
		Factual	300
Gender [98]	-	Male	1,604,068
		Female	1,604,068
Political [98]	-	Democracy	298,961
		Republican	298,961
Offensive [21]	Twitter	Offensive	74,218
		Non-offensive	1,962,224
	Reddit	Offensive	266,785
		Non-offensive	7,096,473

6 RESOURCES AND EVALUATION METHODS

As TST is a relatively new research area, new methods will need to be designed to evaluate the TST algorithms. In this section, we first summarize the downstream tasks and existing datasets used to evaluate the TST models. Next, we discuss the automated and human evaluation methods used to assess the quality of TST algorithms.

6.1 Tasks and Datasets

Datasets. Table 3 summarizes the datasets used in existing studies to evaluate the performance of TST algorithms. These corpora often contain texts labeled with two or more attributes. For example, the Yelp dataset contains review text records labeled with binary sentiment class (i.e., positive or negative), and the Caption dataset contains caption-text records labeled with *romantic*, *humorous*, and *factual* classes. Most of these datasets are non-parallel datasets, i.e., there are no matching text pairs in the different attribute classes, except for Shakespearean-Modern and GYAFC. It is also interesting to note that while the GYAFC corpus is a parallel dataset, most of the existing TST studies assume a non-parallel setting when training the TST models with this dataset.

Many downstream tasks have been proposed to leverage these datasets to evaluate TST models. In the rest of this section, we will review these downstream tasks in greater detail.

Table 4. Sentiment transfer examples.

	Positive Sentiment ⟷ Negative Sentiment
input	Everything is fresh and so delicious!
Ref-0	Everything was so stale.
Ref-1	Everything is rotten and not so delicious.
Ref-2	Everything is stale and horrible.
Ref-3	Everything is stale and tastes bad.

Author Imitation. Author imitation is the task of paraphrasing a sentence to match a specific author's style. To perform this task, Xu et al. [141] collected a parallel dataset, which captures the line-by-line modern paraphrases for 16 of Shakespeare's 36 plays (*Antony & Cleopatra, As You Like It, Comedy of Errors, Hamlet, Henry V, etc.*) using the educational site *Sparknotes* [1]. The goal was to imitate Shakespeare's text style by transferring modern English sentences into Shakespearean style sentences. This dataset is publicly available[2] and was also used in other TST studies [35, 48].

The imitation of author's writing style is an exciting TST task. There are many interesting industrial applications, such as transferring famous novel authors' writing styles into other stories and unifying multiple authors' writing styles to a single author in a collaborative setting. However, the Shakespearean-Modern dataset is the only known corpus that facilitates author imitation in TST studies. There are also some apparent limitations in the corpus; the dataset size is small, and the approach is limited to transfer to only one author's style. An interesting future work may be to collect text written by various authors and transfer the style of text among multiple authors.

Sentiment transfer. Sentiment transfer is a very popular evaluation task adopted in many TST studies. The task involves modifying the sentiment of a sentence while preserving its original contextual content. Table 4 shows an example of the sentiment transfer task. Given the input sentence with positive sentiment, "Everything is fresh and so delicious!", the goal of the TST model is to convert the sentence into negative sentiment while preserving the contextual content information. In this example, the word "Everything" represents the content information and is preserved during the style transfer operation. The examples also reveal an interesting aspect of the sentiment transfer task: while the sentence's style is transferred and the contextual content is preserved, the semantics of the sentence has also changed. For instance, a sentence supporting a particular political party may semantically change to a negative opinion during the sentiment transfer process. Nevertheless, this task is widely used to evaluate TST models, and three popular datasets have been proposed for this task:

- *Yelp*[3] is a corpus of restaurant reviews from Yelp collected in [112]. The original Yelp reviews are on a 5 points rating scale. As part of data preprocessing, reviews with 3 points and above ratings are labeled positive, while those below 3 points are labeled negative. The reviews with an exact 3 points rating are considered neutral and excluded from this dataset.
- *Amazon*[4] is product review dataset from amazon collected in [36]. It is preprocessed using the same method in the Yelp dataset.
- *IMDb*[5] is a movie reviews dataset. Dai et al. [17] constructed this dataset by performing similar data preprocessing methods on a publicly available and popular movie reviews dataset [80].

[1] www.sparknotes.com
[2] http://tinyurl.com/ycdd3v6h
[3] https://github.com/shentianxiao/language-style-transfer
[4] https://github.com/lijuncen/Sentiment-and-Style-Transfer
[5] https://github.com/fastnlp/nlp-dataset

Table 5. Formality transfer examples.

	Informal Formality ⟷ formal Formality
input	He loves you, too, girl...Time will tell.
Ref-0	He loves you as well, but only time can tell what will happen.
Ref-1	He loves you too, lady...time will tell.
Ref-2	He loves you, as well. Time will tell.
Ref-3	He loves you too and time will tell.

Formality transfer. The formality transfer task involves modifying the formality of a given sentence. Typically, an informal sentence is transformed to its formal form and vice versa. Formality transfer is presumably more complex than sentiment transfer, as multiple attributes may affect the formality of text. For instance, the modification of the sentence structure, the length of text, punctuation, and capitalization, may influence text formality. Table 5 shows an example of transferring an informal sentence to its formal form. We illustrate that simple keyword replacement methods cannot achieve the formality transfer of a sentence from the four transferred formal sentences. After an informal sentence is transferred to its formal form, the length of the sentence (as shown in Ref-0) and punctuation may be changed (e.g., replacement of ellipsis with a full stop). Furthermore, unlike sentiments, a sentence's formality is highly subjective; individuals may perceive a sentence's degree of formality differently.

GYAFC[6] is the largest human-labeled parallel dataset proposed for the formality transfer task [103]. The authors extracted informal sentences from Entertainment&Music (E&M) and Family&Relationship (F&R) domains of the Yahoo Answers L6 corpus [7]. The collected dataset was further preprocessed to remove sentences that are too short or long. Finally, the authors crowd-sourced workers to manually re-write the informal sentences to their formal form, resulting in a parallel formality dataset. This dataset was also widely used to evaluate the recent TST models.

Paper-news title transfer. Paper-news title transfer was the task of transferring a title to a different type while preserving the content. Fu et al. [27] collected the *PNTD*[8] dataset, which consisted of paper titles retrieved from academic publication archive websites such ACM Digital Library, Arxiv, Sprint, ScienceDirect, and Nature, and news titles from the UC Irvine Machine Learning Repository.

Captions style transfer. Li et al. 2018 [72] proposed the task of transferring factual formal captions into romantic and humorous styles. The researchers collected the caption dataset[4], where each sentence was labeled as factual, romantic, or humorous. This is also the smallest TST dataset.

Gender style transfer. The differences between male and female writing styles is a widely studied research topic in sociolinguistics. Prabhumoye et al. [98] extended the sociolinguistics studies to perform TST between text written by the different gender, i.e., transfer a text written by a male to female writing style and vice versa. The researchers constructed the gender dataset[9] by first preprocessing a Yelp review dataset annotated with the gender of the reviewers [104] by split the reviews into sentences and preserve the gender label for each sentence. The sentences that are deemed to be gender-neutral are removed from the dataset.

Political slant transfer. Political slant transfer is the task of modifying the writer's political affiliation writing style while preserving the content. Prabhumoye et al. [98] collected comments of Facebook posts from 412 members of the United States Senate and House who have public Facebook pages. The comments are annotated with the

[6]https://github.com/raosudha89/ GYAFC-corpus
[7]https://webscope.sandbox.yahoo.com/catalog.php?datatype=l
[8]https://github.com/fuzhenxin/textstyletransferdata
[9]https://github.com/shrimai/Style-Transfer-Through-Back-Translation

Table 6. Political slant examples.

Republican	defund them all, especially when it comes to the illegal immigrants.
Republican	thank u james, praying for all the work u do .
Democracy	on behalf of the hard-working nh public school teachers- thank you!
Democracy	we need more strong voices like yours fighting for gun control .

Table 7. Dataset statistics for multiple-attribute transfer datasets.

Dataset	Sentiment		Gender		Category				
FYelp	Positive	Negative	Male	Female	American	Asian	Bar	Dessert	Mexican
	2,056,132	639,272	1,218,068	1,477,366	904,026	518,370	595,681	431,225	246,102
Amazon	Positive	Negative	-	-	Book	Clothing	Electronics	Movies	Music
	64,251,073	10,944,310	-	-	26,208,872	14,192,554	25,894,877	4,324,913	4,574,167
Social Media Content	Relaxed	Annoyed	Male	Female	age:18-24	age:65+			
	7,682,688	17,823,468	14,501,958	18,463,789	12,628,250	7,629,505			

congressperson's political party affiliation: democracy or republican. Table 6 shows examples of comments collected in the dataset[9].

Offensive language correction. The use of offensive and abusive language is a growing problem in online social media. The offensive language correction task aims to transfer offensive sentences into non-offensive ones. Santos et al. [21] collected posts from Twitter and Reddit. The posts were subsequently classified into "offensive" and "non-offensive" classes using a classifier pre-trained on an annotated offensive language dataset.

Multiple-attribute style transfer. Thus far, the tasks we have discussed involved transferring text between two style attributes. Lai et al. [65] proposed a multiple-attribute style transfer task and collected multi-style attribute datasets based on Yelp and Amazon review datasets. Table 7 summarizes the statistics of three datasets collected in their studies. The goal is to transfer a text with specific multiple style attributes such as sentiments and gender of the author. The specification of multiple attributes makes the TST task more complex and realistic as the text style should be multi-faceted. For instance, the gender and sentiment of the author could both affect the style of the text. We postulate that multiple-attribute style transfer may be one of the TST research's future directions, and we will discuss this further in Section 9.

6.2 Automated Evaluation

Several automated evaluation metrics have been proposed to measure the effectiveness of TST models [84, 92–94]. Broadly, these metrics evaluate the TST algorithms in three criteria:

(1) The ability to transfer the text style.
(2) The amount of original content preserved after the TST operation.
(3) The fluency of the transferred style sentence.

A TST algorithm underperforming in any of these three criteria is considered ineffective in performing the TST task. For example, a TST algorithm transfers a negative sentiment sentence, "the pasta tastes bad!", to a positive one, "the movie is great!". While the algorithm can transfer the input text style, i.e., from negative to positive sentiment, it fails to preserve the original statement's content, i.e., describing the pasta. Like many other natural language generation tasks, the transferred sentence will also have to achieve a certain level of fluency for the TST algorithm to be useful in real-world applications. Therefore, an effective TST algorithm will have to perform well in all three criteria of the evaluation.

Transfer strength. The transfer strength of a TST model or its ability to transfer text style is commonly measured using *Style Transfer Accuracy* [27, 41, 52, 78, 112]. Typically, a binary style classifier TextCNN [86] is first pre-trained separately to predict the style label of the input sentence. The style classifier is then used to approximate the style transfer accuracy of the transferred style sentences by considering the target style as the ground truth. It is also important to note that the style classifier is not perfect. For instance, when pre-trained on the Yelp and GYAFC datasets and applied to classify tweets on their respective validation dataset, the style classifier can achieve 97.2% accuracy on Yelp dataset, while it is only able to achieve 83.4% accuracy on GYAFC dataset. An alternative metric is to measure the *Earth Mover's Distance [105] between the style distributions of the input text and the transferred text. The Earth Mover's distance can be interpreted as a "cost" to turn one distribution into the other, or how "intense" the transfer is [84].*

Content preservation. To quantitatively measure the amount of original content preserved after the style transfer operation, TST studies have borrowed three automated evaluation metrics that are commonly used in other natural language generation tasks:

- *BLEU*: The BLEU score [95] was originally designed to evaluate the quality of a machine-translated text. The BLEU score was one of the first metrics to claim a high correlation with human judgment on the translated text quality. To compute the BLEU score, the machine-translated text is compared with a set of good-quality reference translations. Similarly, in TST, the BLEU is computed when parallel TST datasets or human references are available. Specifically, we compute the BLEU score between the transferred sentences and the parallel references available to evaluate content preservation.
- *source-BLEU (sBLEU)*: Nevertheless, most of the TST tasks assume a non-parallel setting, and matching references of style transferred sentences are not always available. Therefore, TST studies often apply a modified *source-BLEU (sBLEU)* score where the transferred sentence is compared with its source input sentence. The intuition is that the content is assumed to be preserved, and the transferred sentence would share many overlap n-grams with the original sentence.
- *Cosine Similarity*: Fu et al. [27] calculated the cosine similarity between original sentence embeddings and transferred sentence embeddings. The intuition is that the embeddings of the two sentences should be close to preserve the semantics of the transferred sentence.
- *Word Overlap*: Vineet et al. [52] argued that the cosine similarity is not a sensitive metric as the original and transferred sentences may have high cosine similarity scores even the content of the sentences are different. Thus, they employed a simple metric that counts the unigram word overlap rate of the original and style transferred sentences. Noted that stop words and style-attributed words (e.g., sentiment words) are excluded in the word overlap calculation.

Fluency. Generating fluent sentences is a common goal for almost all natural language generation models. A common approach to measuring a sentence's fluency is using a trigram Kneser-Ney language model [62]. The Kneser-Ney language model is pre-trained to estimate the empirical distribution of trigrams in a training corpus. Subsequently, the *perplexity score* of a generated sentence is calculated by comparing the sentence's trigram and the estimated trigram distribution. The intuition is that a generated sentence with a lower perplexity score is considered more "aligned" to the training corpus and, therefore, considered more fluent. In TST tasks, the language model is similarly trained on the TST datasets, and the perplexity scores of the style transferred sentences are computed to evaluate the sentences' fluency.

Table 8. Dataset statistics for Yelp and GYAFC.

Dataset	Subset	Attributes	Train	Dev	Test
Yelp	-	Positive	267,314	38,205	76,392
		Negative	176,787	25,278	50,278
GYAFC	F&R	Informal	51,967	2,788	1,332
		Formal	51,967	2,247	1,019

6.3 Human Evaluation

Few TST studies have performed human evaluations on their proposed TST algorithms [72, 112] as such evaluations are often expensive and laborious. Human workers are crowd-sourced in a typical human evaluation setting to rate how the style transferred sentence fair on the three evaluation criteria using a range scale. For example, given a pair of original and transferred sentences, a human worker is asked to rate how well the content is preserved in the transferred sentence on a scale of 1 to 5 points, with 5 points being "*very well preserved*". Multiple human workers are asked to evaluate a given pair of original and transferred sentences, and the average scores are reported to reduce individual bias. Although researchers have put in great effort to ensure the quality of the human evaluation on TST tasks, the evaluation approach has proven to be very challenging as the interpretation of the text style is subjective and may vary across individuals [84, 92, 93]. Nevertheless, human evaluations still offer valuable insights into how well TST algorithms are able to transfer style and generate sentences that are acceptable by human standards.

7 REPRODUCIBILITY STUDY

Although most of the existing TST methods were evaluated in the original works using the downstream tasks discussed in Section 6, the experiments were often carried out with no or few baselines. Thus, we conduct a reproducibility study[10] and benchmark 19 TST models on two popular corpora: Yelp reviews and GYAFC, representing the sentiment transfer task and formality transfer task, respectively. To the best of our knowledge, this is the first time where so many TST models are evaluated on the same datasets. Specifically, the experimental results from this study provide new insights into how each TST algorithm fares against each other in terms of *transfer strength*, *content preservation*, and *fluency*. This section is organized as follows: We first describe the experimental setup of our reproducibility study. Next we discuss the experimental results on *sentiment transfer* and *formality transfer* tasks. We also perform the trade-off analyses to investigate how the relationships between multiple evaluation criteria influence the TST model performance. Finally, we perform human evaluation on a subset of representative TST models and report the results.

7.1 Experimental Setup

Environment Settings. The experiments were performed on an Ubuntu 18.04.4 LTS system with 24 cores, 128 GB RAM, and a clock speed of 2.9 GHz. The GPU used for deep neural network-based models was Nvidia GTX 2080Ti. We followed the environmental requirements and hyperparameter settings of the released code implementations of the TST models to reproduce the experimental results. Table 8 shows the training, validation, and test splits of the Yelp and GYAFC datasets used in our experiments.

Evaluation Metrics. We adopt the evaluation metrics discussed in Section 6 to measure the performance of the TST models. Specifically, we apply the Style Transfer Accuracy (*ACC*) to measure transfer strength. For measuring

[10] Code implementation of the reproduced model are compiled in this repository: https://gitlab.com/bottle_shop/style/tst_survey

content preservation, we adopt *BLEU*, *sBLEU*, Cosine Similarity (*CS*), and Word Overlap (*WO*). Noted that the *BLEU* score is only computed for the GYAFC dataset as the human references of the sentences in test set are available. We compute the perplexity score (*PPL*) to quantify the fluency of the transferred sentences. In practice, we use the script **multi-bleu.perl [11] to calculate *BLEU* and *sBLEU* scores. For *WO*, we exclude both stopwords and sentiment words [12].** Finally, we compute two average metrics that consider all evaluation aspects:

- Geometric Mean (*G-Score*): We compute the geometric mean of *ACC* (transfer strength), *sBLEU* (content preservation), *WO* (content preservation), and *1/PPL* (fluency). We excluded the *CS* measure in mean computation due to its insensitivity, and we take the inverse of the calculated perplexity score because a smaller *PPL* score corresponds to better fluency.
- Harmonic Mean (*H-Score*): Different averaging methods reflect different priorities. Thus, we also compute the harmonic mean of *ACC*, *sBLEU*, *WO*, and *1/PPL*.

Reproduced Models. We limit our reproducibility study to these 19 TST models as they had published their implementation codes. We hope to encourage fellow researchers to publish their codes and datasets as it can promote this field's development. We have also grouped and color-coded the models according to the strategies discussed in Section 5. Orange represents the explicit style-content disentanglement methods, green denotes methods that applied implicit disentanglement strategy, and blue represents methods that performed TST without style-content disentanglement. Specifically, we reproduced and implemented the following TST models:

Explicit Style-Content Disentanglement 5.3.

- **DeleteOnly; Template; Del&Retri** [72]: A style keyword replacement method, which disentangles the style and content of sentence explicitly by keyword replacement. The authors proposed three variants of their mode: DeleteOnly, which first remove the removed style attributed keywords from the source sentence. Subsequently, the source sentence's latent representation is combined with the target style attribute and input into a sequence model to generate the sentence in the target style. The Template model is simply replacing the deleted style attributed keywords with target style keywords. The Del&Retri model first performed the same keyword removal as DeleteOnly method. Next, it retrieves a new sentence associate with the target attribute. Lastly, the keyword-removed source sentence and the retrieved sentence are input into a sequence model to generate the transferred style sentence.
- **B-GST; G-GST** [117]: A TST model that extended the work in [72] and proposed the Generative Style Transformer (GST) to perform text style transfer. There are two variants of the GST model: the Blind Generative Style Transformer (B-GST) and the Guided Generative Style Transformer (G-GST).
- **PTO** [132]: A style keyword replacement TST model that applied reinforcement-learning to hierarchically reinforced a *Point-Then-Operate* (PTO) sequence operation. The PTO operation has two agents: a high-level agent that iteratively proposes operation positions and a low-level agent that alters the sentence based on the high-level proposals. Using this reinforcement framework, the style-attributed keywords are replaced explicitly to perform TST.
- **UST** [137]: A style keyword replacement method, which utilized cycled reinforcement learning to iteratively replace style attributed keywords while maintaining the content in text for text style transfer. This model was originally implemented for the sentiment transfer task.

[11] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl
[12] https://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

- **SMAE** [149]: A style keyword replacement model to perform TST by disentangling style and content explicitly. The model was originally designed for sentiment transfer. The sentiment attribute words are first detected, and a sentiment-memory-based auto-encoder model is subsequently used to perform sentiment modification without parallel data.

  All the explicit style-content disentanglement methods are discussed in detail in Section 5.3.

Implicit Style-Content Disentanglement

- **DRLST** [52]: An adversarial learning TST model that incorporates auxiliary multi-task and adversarial objectives for style prediction and bag-of-word prediction respectively to perform text style transfer. This method is discussed in Section 5.4.
- **BST** [98]: A back-translation based TST model that employed a pre-trained back translation model to rephrase a sentence while reducing its stylistic characteristics. Subsequently, separate style-specific decoders were used for style transfer. This method is discussed in Section 5.5.
- **CAAE** [112]: An adversarial learning TST model that implicitly disentangle the text's style. Specifically, the model assumes a shared latent content distribution across different text corpora and proposes a method that leverages refined alignment of latent representations to perform TST. This method is discussed in Section 5.4.
- **Ctrl-Gen** [41]: An attribute-controlled TST model that used variational auto-encoders and style classifier to guide the learning of a style attribute to control the generation of text in different styles. This method is discussed in Section 5.6.1.
- **ARAE** [151]: A generic natural language generation technique that utilizes adversarial learning to modify the specific attributes in text. TST is one of the model's applications proposed in its original paper. This method is discussed in Section 5.4.
- **Multi-Dec; Style-Emb** [27]: A adversarial learning TST model that utilized a style classifier to perform disentanglement of style and content representation for style transfer task. Two variants of the model were proposed: The multi-decoder (Multi-Dec) model that uses different decoders to generate text with different styles. The style embedding (Style-Emb) model concatenates the style embedding vector with content representation to generate different style text with one decoder. This method is discussed in Section 5.4.

Without Style-Content Disentanglement

- **DualRL** [78]: A reinforcement-learning based-TST model that utilized two seq2seq models to transfer between two text styles. Specifically, this model considers the learning of source-to-target style and target-to-source style as a dual-task that mutually reinforce each other to perform TST without disentangling style and content. This method is discussed in Section 5.8.
- **DAST; DAST-C** [70]: An attribute-controlled TST model that performs TST in a domain-aware manner. Two variants are proposed: The Domain Adaptation Style (DAST) model and DAST with generic content information (DAST-C). In these models, latent style attributes and domain vectors are learned to perform TST across domains. This method is discussed in Section 5.6.2.
- **PFST** [35]: A probabilistic deep generative TST model. The model combines a language model prior and an encoder-decoder transducer to infer the sentence's latent representations in an assumed parallel style corpus. The inferred latent representations are subsequently used to generate the sentence of a specific style via a decoder. This method is discussed in Section 5.10.

Table 9. TST results on Yelp dataset (sentiment transfer task).

Model	*ACC(%)*	*sBLEU*	*CS*	*WO*	*PPL*	*G-Score*	*H-Score*	#Params	FLOPs
DeleteOnly	84.2	28.7	0.893	0.501	115	1.80	0.034	9.18M	27.98G
Template	78.2	48.1	0.850	0.603	1959	1.04	0.002	-	-
Del&Retri	88.1	30.0	0.897	0.464	101	1.87	0.039	9.88M	29.78G
B-GST	89.2	46.5	0.959	0.649	216	1.88	0.018	-	-
G-GST	72.7	52.0	0.967	0.617	407	1.55	0.010	-	-
PTO	82.3	57.4	**0.982**	0.737	245	1.94	0.016	-	-
UST	74.0	41.0	0.929	0.448	394	1.36	0.010	79.97M	2390.6G
SMAE	84.4	14.8	0.907	0.294	210	1.315	0.019	76.27M	4418.5G
DRLST	91.2	7.6	0.904	0.484	**86**	1.41	**0.045**	64.0M	0.149G
BST	83.1	2.3	0.827	0.076	261	0.49	0.015	-	-
CAAE	82.7	11.2	0.901	0.277	145	1.15	0.027	12.92M	0.436G
ARAE	83.2	18.0	0.874	0.270	138	1.31	0.028	4.20M	5.61G
Ctrl-Gen	89.6	49.5	0.953	0.707	384	1.69	0.010	13.7M	0.153G
Multi-Dec	69.6	17.2	0.887	0.244	299	0.99	0.013	-	-
Style-Emb	47.5	31.4	0.926	0.433	217	1.31	0.018	-	-
DualRL	79.0	**58.3**	0.97	**0.801**	134	**2.29**	0.030	44M	0.862G
DAST	90.7	49.7	0.961	0.705	323	1.77	0.012	26.68M	0.822G
DAST-C	**93.6**	41.2	0.933	0.560	450	1.48	0.009	24.51M	0.508G
PFST	85.3	41.7	0.902	0.527	104	2.06	0.038	34.32M	76.19G

7.2 Sentiment Transfer

Table 9 shows the performance of various TST models on the sentiment transfer task. We observe that there are no TST models that achieved the best performance in all evaluation metrics. DualRL, PTO, B-GST, and PFST have achieved a well-balanced trade-off among between text fluency, content preservation, and the style transfer accuracy. DRLST has achieved the second best transfer accuracy. However, the model also suffers a very low *sBLEU* score, suggesting the DRLST's ineffectiveness in preserving the content of the original sentence. Moreover, we observe that most of the implicit style-content disentanglement methods have poor performance on content preservation. A potential reason could be that some content information may be lost while performing disentanglement of style and content. We also note that the different average methods, i.e., *G-Score* and *H-Score*, weighted the different evaluation metrics differently. For instance, the *H-score* gives a higher weight to perplexity scores of generated sentences. Thus, DRLST, which has the lowest *PPL* score, also has the highest *H-score*. Conversely, the Template modelhas the highest *PPL* score and lowest *H-score*.

More interestingly, we observed that the style keyword replacement methods such as DeleteOnly, Template, Del&Retri, B-GST, G-GST, UST, PTO, and SMAE, have achieved good performance in the sentiment transfer tasks. The methods have achieved a high transfer accuracy score while preserving the content information, i.e., high *sBLEU*, *CS*, and *WO* scores. A possible reason for the style keyword replacement methods good performance might be due to the nature of the task; the sentiment of a sentence can be easily modified by replacing keywords related to the source sentiment. For example, replacing "fresh" with "rotten" would transform the sentence from positive to negative sentiment. However, it is interesting to note that the Template method, which is an algorithm that simply replaces the sentiment-related keywords, has a high perplexity score, which indicates bad performance in sentence fluency. This motivates more complex generative approaches that can prevent the generation of implausible sentences by simple keyword replacement.

7.3 Formality Transfer

Table 10 shows the performance of various TST models on the formality transfer task. Similar to the observation in sentiment transfer, none of the TST models is able to score well on all evaluation metrics. We noted that the average style transfer accuracy in GYAFC is 52.9%, which is significantly lower than Yelp's average score of 84.4%. This highlights the difficulty of the formality transfer task. We also observe that most models performed worse in this task compared to the sentiment transfer task. It is also unsurprising that the style keyword replacement methods did not perform well in the formality transfer task; most of these models achieved a low style transfer accuracy. Some of the adversarial learning-based TST models, such as CAAE [112] and DRLST [52], had achieved high style transfer accuracy but very low content preservation as these models lack the mechanism to control content preservation during the generative process. Interestingly, we observe that the attribute-controlled TST methods, i.e., Ctrl-Gen [41], DAST [70], and DAST-C [70] have achieved good performance both style transfer accuracy and content preservation. Similar to sentiment transfer, we observe that the implicit style-content disentanglement methods, i.e., DRLST, BST, CAAE, ARAE, and Multi-Dec, also performed poorly on content preservation for the formality transfer task. Specifically, the scores of content preservation metrics are even lower than that in the sentiment transfer task. The methods performing TST tasks without style-content disentanglement have achieved the best performance for the formality transfer task.

The GYAFC dataset provided the performance of four human references performing the formality transfer task on the test dataset (as shown in the bottom of Table 10). On average, the human references had achieved 78.1% style transfer accuracy. This is considered a reasonable performance, given that the pre-trained binary classifier only managed to achieved 83.4% accuracy on the test set. Furthermore, text formality is subjective, and the four human references may have different opinions on the degree of formality in text.

As the GYAFC dataset is a parallel dataset, i.e., there are matching sentences in source and target styles, we are able to compute the *BLEU* score between the transferred style sentence and the matching sentence in the target style. Unsurprisingly, the human references have achieved the highest *BLEU* score, suggesting that the sentences generated by the human references are quite similar to the matching sentences in the target style. In comparison, the TST models fair poorly in the *BLEU* scores. We also observe that the TST models' average content preservation metrics scores in the formality transfer task are lower than the scores in the sentiment transfer task. For instance, the *WO* scores in the sentiment transfer task are higher because only a few keywords need to be replaced to perform the style transfer. However, in the formality transfer case, a more drastic and complex modification of the text has to be performed for the style transfer. As such, there will be less word overlap between the original sentence and the transferred sentence, resulting in a lower *WO* score. The limitation of existing metrics in measuring content preservation in formality transfer highlights the need to search for better evaluation methods for this challenging task.

7.4 Computational Efficiency Evaluation

We have also reported the computational resources needed to train the TST models in Table 9 and 10. Specifically, we compute the number of parameters textit#Params and the floating point operations *FLOPs* of the various TST models. For Tensorflow framework-based models, we use **tf.trainable_variables()** function to compute the number of parameters, and **tf.profiler.profile()** function to compute the *FLOPs*. For Pytorch framework-based models, we use a third-party library name THOP [13] to computer #Params and FLOPs. Note that the computed #Params and FLOPs are approximations as the built-in functions of Tensorflow and THOP are not able to evaluate the custom functions defined

[13] https://github.com/Lyken17/pytorch-OpCounter

Table 10. TST results on GYAFC dataset (formality transfer task).

Model	*ACC(%)*	*sBLEU*	*hBLEU*	*CS*	*WO*	*PPL*	*G-Score*	*H-Score*	#Params	FLOPs
DeleteOnly	26.0	35.4	16.2	0.945	0.431	82	1.48	**0.047**	9.19M	27.89G
Template	51.5	45.1	19.0	0.943	0.509	111	1.81	0.035	-	-
Del&Retri	50.6	22.1	11.8	0.934	0.345	94	1.42	0.041	9.99M	29.78G
B-GST	30.3	22.5	11.6	**0.951**	**0.557**	117	1.34	0.034	-	-
G-GST	31.0	20.7	10.2	0.941	0.556	127	1.29	0.031	-	-
UST	23.6	0.5	0.5	0.881	0.012	**28**	0.27	0.035	79.97M	2390.6G
SMAE	21.6	6.5	1.2	0.898	0.079	74	0.62	0.046	76.27M	4418.5G
DRLST	71.1	4.2	2.7	0.909	0.342	86	1.04	0.045	64.0M	0.149G
BST	69.7	0.5	0.5	0.883	0.04	69	0.38	0.042	-	-
CAAE	72.3	1.8	1.5	0.896	0.028	55	0.51	0.044	12.92M	0.436G
ARAE	76.2	4.8	2.2	0.903	0.042	77	0.67	0.040	4.20M	5.61G
Ctrl-Gen	73.1	57.0	15.6	0.943	0.446	168	1.82	0.023	13.7M	0.153G
Multi-Dec	22.2	13.4	5.9	0.911	0.168	146	0.76	0.026	-	-
Style-Emb	27.7	8.3	3.6	0.897	0.102	136	0.64	0.027	-	-
DualRL	56.7	**61.6**	18.8	0.944	0.447	122	**1.89**	0.032	44M	0.862G
DAST	73.1	50.6	14.3	0.934	0.350	204	1.59	0.019	26.68M	0.822G
DAST-C	78.2	48.5	13.8	0.927	0.328	308	1.42	0.013	24.51M	0.508G
PFST	50.8	55.3	16.5	0.940	0.466	200	0.51	0.020	34.32M	76.19G
Human0	78.1	20.5	43.5	0.942	0.393	80	1.67	0.048	-	-
Human1	**78.7**	18.2	43.2	0.931	0.342	199	1.25	0.020	-	-
Human2	78.2	18.6	43.4	0.932	0.354	192	1.28	0.021	-	-
Human3	77.4	18.8	43.5	0.931	0.354	196	1.27	0.020	-	-

by model developers. Furthermore, we are not able to report the metrics for some of the TST models due to various technical constraints. For instance, orangeTemplate is rule-based model, B-GST and G-GST use pre-trained language model GPT [101], and the Pytorch version of BST is too old. We are also not able to retrieve the metrics for Multi-Dec and Style-Emb as the models utilized the deep learning framework Theano.

For TST methods that explicitly disentangled style and content, we observed that DeleteOnly and Del&Retri have relatively lower #Params and FLOPs compared to .UST and SMAE. This could be attributed to the technique used to perform the explicit style and content disentanglement. DeleteOnly and Del&Retri have utilized a token frequency-based method to remove style-attributed keywords in a given sentence before performing keyword replacement [72]. On the other hand, UST and SMAE trained a LSTM model with attention mechanism to identify and remove style-attributed keywords in a sentence. The addition of sequence models to perform the style-attributed keyword removal process added more #Params needed to train the TST two models. DRLST has a relatively higher #Params compared to the other TST methods that performed TST methods that implicitly disentangled style and content. This high #Params could be attributed to the multi-task learning objectives in the greenDRLST [52]. DualRL has the highest #Params among methods that perform TST without style-content disentanglement because of the complex neural network architecture; the model needs to learn and map two Seq2Seq models to perform TSTs [78].

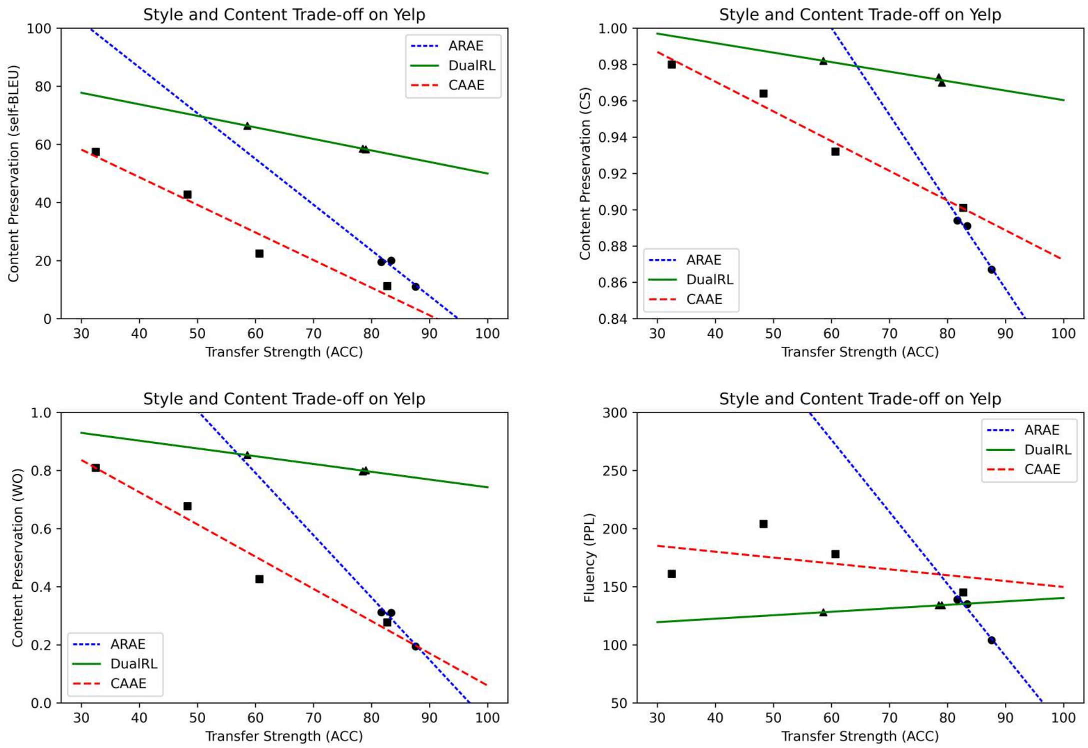

Fig. 10. Metrics trade-off analysis for sentiment transfer on Yelp review dataset

7.5 Evaluation Metrics Trade-off Analysis

Besides evaluating the TST models on the two style transfer tasks, we also reproduced the evaluation metrics trade-off analysis proposed in [84]. Specifically, we create variants of TST models by varying their hyperparameters and studying the trade-off effects between pairs of evaluation metrics. We select ARAE, DualRL, and CAAE models for our analysis.

The trade-off between style transfer accuracy and content preservation has been discussed in existing TST studies [27, 84]. For instance, Fu et al. [27] performed the trade-off analysis between style transfer accuracy and content preservation of Multi-Dec and Style-Emb for the sentiment transfer task. Similar experiments were also conducted in [84] to evaluate the trade-off between content preservation, style transfer accuracy, and fluency of CAAE, ARAE, and Del&Retri models for the sentiment transfer task. However, there is no known study that perform the trade-off analysis for the formality transfer task. The section aims to fill this gap by performing and comparing the trade-off analysis on the sentiment transfer and formality transfer tasks.

Fig. 10 show the trade-off analysis results on sentiment transfer task. Specifically, Fig. 10A,B and C show the trade-off relationships between style transfer strength and content preservation metrics, while Fig. 10D shows the trade-off relationship between style transfer strength and fluency metric. Similar to the observations made in [84], we notice that as the transfer strength of increases, content preservation metrics decrease across the three models. However, the trade-off relationship between the transfer strength and sentence fluency is less obvious as we notice that ARAE is able to achieve lower *PPL* when *ACC* increases. Similar observations are made for the formality transfer task in Fig. 11.

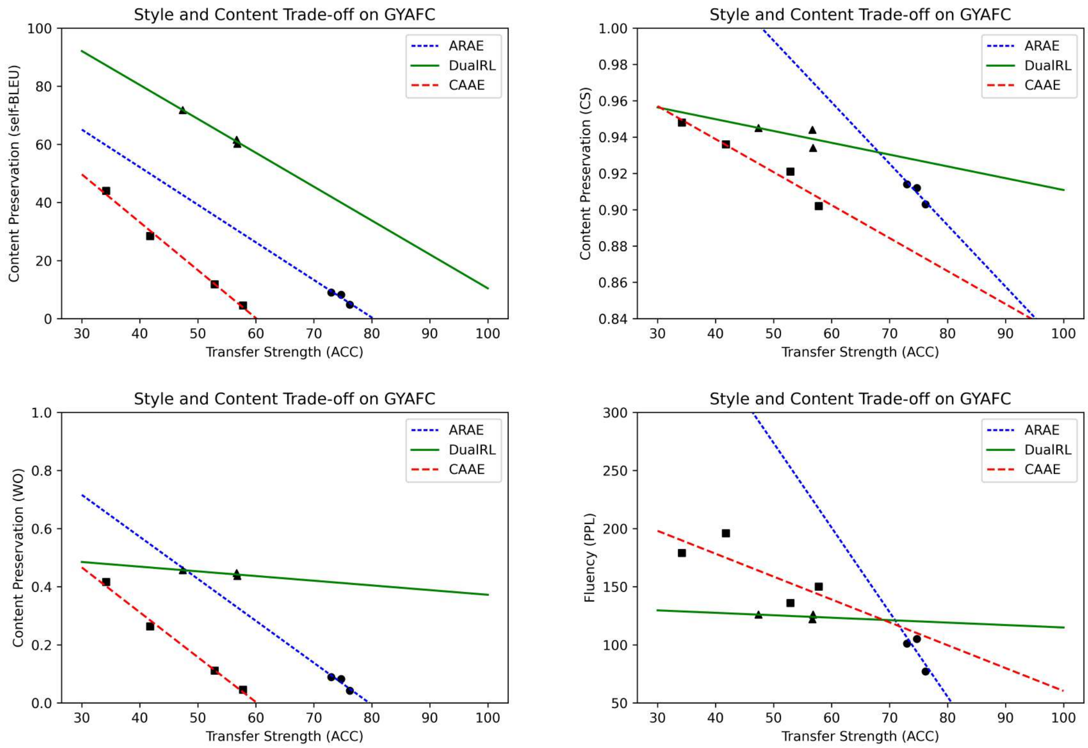

Fig. 11. Metrics trade-off analysis for formality transfer on GYAFC review dataset

Interesting, we noted that for the formality transfer task, even when content preservation is reduced to the lowest, it is still not able to achieve a high style transfer accuracy.

The observations made in our trade-off analysis is consistent to the previous to the observations made in previous studies [27, 84]. Specifically, when transfer strength scores increase, content preservation scores decrease, and vice versa. A potential reason for observation might be due to the entanglement between semantic and stylistic properties in natural language; it is hard to separate the two properties, and changing one affects the other. Therefore, when optimizing to transfer the style in text, it is hard to maintain the sentence's semantic, i.e., the content information.

7.6 Human Evaluation

To further evaluate the TST models' performance, we conducted a human-based evaluation study. We focus our human evaluation on representative models from the three TST strategies, and the models are evaluated on sentiment transfer and formality transfer tasks.

For the sentiment transfer task, we first randomly sampled 50 negative and 50 positive sentences from the Yelp dataset. Next, we perform TST for the sampled sentences using PTO, DRLST and DualRL, which are the best performing models from the various TST strategies when evaluated using *G-Score*. Similarly, for the formality transfer task, we randomly sampled 50 formal and 50 informal sentences from the GYAFC dataset. Subsequently, we perform TST for the

Table 11. Human evaluation results of selected TST models on Yelp and GYAFC datasets.

	Yelp					GYAFC				
Model	*Style*	*Content*	*Fluency*	*G-Score*	*H-Score*	*Style*	*Content*	*Fluency*	*G-Score*	*H-Score*
PTO	3.88	3.06	3.24	3.37	3.35	-	-	-	-	-
Template	-	-	-	-	-	3.03	**3.74**	3.29	3.34	3.33
DRLST	**4.26**	2.27	4.22	3.44	3.29	3.77	1.86	**3.94**	3.02	2.84
DualRL	3.84	**3.59**	**4.26**	**3.88**	**3.87**	**3.81**	3.55	3.58	**3.64**	**3.64**

sampled sentences using orangeTemplate, DRLST and DualRL, which are the best performing representative models for the formality transfer task.

We recruited six linguistic researchers (i.e., evaluators) to evaluate the style-transferred sentences generated by the TST models. Each evaluator is assigned to evaluate pairs of sentences: the source sentence and the style-transferred sentence. The evaluators were asked to evaluate the style-transferred sentences on the three criteria discussed in the earlier section:

(1) **Transfer Strength:** The evaluators were asked to rate the style of the transferred sentences using a 5-point Likert scale. 1: strongly opined that the sentence is in source style (e.g., positive sentiment), and 5: strongly opined that the sentence is in target style (e.g., negative sentiment).
(2) **Content Preservation:** The evaluators were asked to compare the given source sentence and transferred sentence and rate the amount of content preserved in the transferred sentence using a 5-point Likert scale. 1: source and transferred sentences cover absolutely different content, and 5: source and transferred sentences cover absolutely same content.
(3) **Fluency:** The evaluators were asked to rate the fluency of the transferred sentences using a 5 Likert scale. 1: unreadable with too many grammatical errors, 5: perfect and fluent sentence.

To minimize biases, we randomly assign every style-transferred sentence to two evaluators, and the models' names will not be displayed to evaluators.

Table 11 shows the results of our human evaluation experiment. We compute the models' average 5-point Likert scores for transfer strength, content preservation, and fluency criteria. We also report the geometric mean score (*G-Score*) and harmonic mean score (*H-Score*) of the three evaluated criteria.

Examining the results of the sentiment transfer task on the Yelp dataset, we observe that the DualRL model can achieve the best and balanced performance. This demonstrated DualRL's ability to generate fluent sentences in target-style while keeping a good balance between style transfer strength and content preservation. DRLST is observed to achieve the best performance on transfer strength. However, the model has performed poorly on content preservation, suggesting that the model can generate sentences in target-style, but the content differs significantly from the source sentence. This observation is consistent with automatic evaluation as DRLST has a low *sBLEU* score. The explicit style keyword replacement model PTO can achieve competitive performance on transfer strength and content preservation but can not generate fluent sentences as the other two models.

For the formality transfer task on GYAFC dataset, we observed that the orangeTemplate had achieved the best content preservation among compared TST models. However, the model has the worse performance on transfer strength. Similar to sentiment transfer task, DualRL achieved the best performance on transfer strength, *G-Score*, and *H-Score*. DRLST has reasonable performance on transfer strength and fluency but is unable to preserve content information during the TST process.

8 ETHICAL CONSIDERATIONS

With the recent increase of attention on artificial intelligence research's ethical issues, it is pertinent to discuss TST research's ethical considerations. This section highlights the potential ethical issues of TST applications and offers some guidelines to avoid ethical misconduct in future TST research.

8.1 Negative Use-Cases

The natural language processing (NLP) community has initiated thought-provoking discussion on the ethical issues of NLP research and applications. Hovy and Spruit [37] discussed the social implications of NLP technology and research and underlined their ethical significance. The ethical discussion on other NLP technology is also applicable to TST research. For instance, TST researchers need to be wary that the TST techniques created can have unintended negative consequences. While TST techniques can be applied to many exciting and beneficial settings mentioned in Section 4, there are potential negative use-cases if the developed TST techniques are misused.

Content Manipulation and Forgery. TST method could be misused to perform malicious content manipulation and forgery. For instance, TST methods can be applied to manipulate the polarity of reviews, such as Yelp and Amazon reviews; fraudulent sellers may apply the TST method to transfer negative reviews on their products to generate fake positive reviews and generate negative views on their competitors. Similarly, malicious users can apply TST techniques to forge documents and content based on specific user's writing styles. This post challenges to forensic investigation efforts attempt to detect document forgery [5, 40, 44].

Social Bots and Sock-Puppeting. TST methods could also be applied in social media. For instance, the political slant transfer task [98] may raise ethical issues if applied in social bots to generate social media posts advocating certain political ideas and manipulating the political views of the mass. Similarly, TST could also enable sock-puppeting or creating an army of social bots with different persuasive styles to promote anti-social behaviors such as online hate speeches [25] or cyberbullying against individuals or groups [107].

8.2 Ethical Guidelines

Researchers and companies need to raise awareness of these potential misuses of the developed TST techniques. Specifically, companies and research institutes should consider including an ethical review process to examine the TST research and development of TST applications. Leidner and Plachouras [69] proposed a framework to encourage organizations to build NLP applications with ethical considerations, and this framework could also be applied to TST research. Specifically, companies should set up an Ethics Review Board (ERB) to examine the ethical issues before, during, and after executing the TST research and development. Most research institutes and universities have ERBs, and we encourage the ERBs to include artificial intelligence and NLP experts when examining the ethical issues in the proposed TST research. By having ethics stakeholders participating in various research and development stages, we hope that TST researchers could build-in ethical considerations in their studies and mitigate the ethical concerns.

9 FUTURE RESEARCH DIRECTION AND OPEN ISSUES

TST is a relatively new research area. While existing works have established a foundation for TST research, this section outlines several promising prospective research directions. We also discuss the open issues that we believe are critical to the present state of the field.

9.1 Deeper Dive into Style-Content Disentanglement

As discussed in Section 5.1.2, text style and content disentanglement remains a open research questions. Researchers who intend to propose new TST methods that perform style and content disentanglement need to design new experiments to demonstrate or quantify the extent of the disentanglement. The style and content disentanglement should also be investigated in different TST tasks, e.g., sentiment and formality transfer tasks.

More studies could also be done on the style embeddings learned by the existing techniques. Currently, we have little understanding on the style representations learned using existing techniques; besides knowing the style representation supposedly has some correlation with the style labels, we do not know much about the information that are preserved in the style representations. For instance, in learning the style representations for the formality transfer tasks, little is known about the preservation of the sentence structure in the representations, and the sentence structure may have an impact on the formality of the text. To this end, a potential future research direction would be to conduct a deeper analysis of the style representations learned for the different tasks using existing techniques. We believe that this will provide new insights, which can guide the development of future TST techniques.

9.2 Unsupervised Text Style Transfer

While most of the exiting TST methods are developed for the non-parallel dataset setting, these techniques continue to require a large amount of style labeled data to guide the transferrence of text styles. A promising research direction would be to explore unsupervised methods to perform TST with little or no labeled data. For instance, recent studies [31, 46] have explored guiding the transfer of text style by scoring the sentences' semantic relatedness, fluency, and readability grade instead of the style labels. We postulate that more aspects of the text, such as tone, brevity, sentence structures, etc., can be explored to train the future TST models and reduce the dependence on the style labels.

9.3 Going Beyond Transferring Between Two Styles

Currently, most of the existing TST methods focus on transferring the text between two styles. We believe that TST studies should go beyond performing a binary style transfer and explore richer and more dynamic tasks. For example, Lai et al. [65] proposed a multiple-attribute style transfer task where a text is transferred by specifying multiple style attributes such as a sentiment, gender of the author, etc. Domain-aware TST method has also been explored where we consider the domain of the text (e.g., food or movie reviews) when transferring the text styles (e.g., from positive to negative sentiment). We believe that more dynamic TST tasks with better real-life applications will be a promising future research direction.

9.4 Style in other languages

Most of the existing TST models were applied to English corpora, neglecting TST's potential in other languages. Different languages may have their unique text style properties. Therefore, novel TST methods should be designed to capture the language-specific stylistic properties. Mizukami et al. [85] designed a dialogue system that modeled individual Japanese writers' text style using statistical machine translation-based technique. However, there is no recent exploration to perform TST in non-English languages, and we would encourage TST researchers to address this research gap. Moreover, we believe that exploring the style features in different languages may improve our understanding of text style and style representations.

9.5 Automatic Evaluation for Text Style Transfer

Our experimental evaluation in section 7 has illustrated the challenges of evaluating the effectiveness of TST models. The existing evaluation methods have a few limitations. Firstly, the evaluation of text style transfer based on transfer accuracy is limited by the performance of style classifier. Secondly, similar to previous studies [27, 94], we notice that the transfer strength is inversely proportional to the content preservation, suggesting that these metrics may be complementary and challenging to optimize simultaneously. The limitations of existing evaluation metrics motivate the exploration of novel automatic evaluation metrics to evaluate TST models.

10 DISCUSSION AND CONCLUSION

Although TST is a relatively new branch of the natural language processing field, a considerable amount of TST research has been conducted in recent years. The explosive growth of TST research has generated many novel and interesting TST models. This survey aims to organize these novel TST models using a taxonomy (cf. Fig. 1). We also summarize the common techniques used by modern TST models to transfer text styles. We also emphasize the important TST research trends, such as the shift from TST models attempting to disentangle text style from content to aiming to perform TST without any style-content disentanglement. While we postulate that the trend on performing TST without any style-content disentanglement will continue, we believe that the study on style representation remains an interesting research direction that deserves further exploration.

Besides discussing the common TST techniques, we also conducted a large-scale reproducibility study where we replicated and benchmarked 19 state-of-the-art TST algorithms on two publicly available datasets. To the best of our knowledge, this is the first large-scale reproducibility study on TST methods. The results of our study show that none of the TST methods could dominate on all evaluation metrics. This suggests the complexity of the TST task, wherein different methods may have advantages in different aspects, and there is no simple way to declare a winner. The evaluation analysis in our reproducibility study also advocated the need to search for better TST evaluation metrics.

We believe that research on TST will continue to flourish, and the industry will continue to find more exciting applications for the existing TST methods. We hope that this survey can provide readers with a comprehensive understanding of the critical aspects of this field, clarify the important types of TST methods, and shed some light on future studies.

REFERENCES

[1] Sameera A Abdul-Kader and JC Woods. 2015. Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications* 6, 7 (2015).

[2] Vikas Ganjigunte Ashok, Song Feng, and Yejin Choi. 2013. Success with style: Using writing style to predict the success of novels. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1753–1764.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[4] Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 10–21.

[5] Marcelo Luiz Brocardo, Issa Traore, Sherif Saad, and Isaac Woungang. 2013. Authorship verification for short messages using stylometry. In *2013 International Conference on Computer, Information and Telecommunication Systems (CITS)*. IEEE, 1–6.

[6] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Frederick Jelinek, John Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational linguistics* 16, 2 (1990), 79–85.

[7] 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, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[8] Fazli Can and Jon M Patton. 2004. Change of writing style with time. *Computers and the Humanities* 38, 1 (2004), 61–82.

[9] Yixin Cao, Ruihao Shui, Liangming Pan, Min-Yen Kan, Zhiyuan Liu, and Tat-Seng Chua. 2020. Expertise Style Transfer: A New Task Towards Better Communication between Experts and Laymen. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[10] Keith Carlson, Allen Riddell, and Daniel Rockmore. 2018. Evaluating prose style transfer with the Bible. *Royal Society open science* 5, 10 (2018), 171920.

[11] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799* (2020).

[12] Marilyn J Chambliss and Ruth Garner. 1996. Do adults change their minds after reading persuasive text? *Written Communication* 13, 3 (1996), 291–313.

[13] Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. 2018. Adversarial text generation via feature-mover's distance. In *Advances in Neural Information Processing Systems*. 4666–4677.

[14] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731* (2016).

[15] Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving Disentangled Text Representation Learning with Information-Theoretic Guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 7530–7541. https://doi.org/10.18653/v1/2020.acl-main.673

[16] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.

[17] 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*. 5997–6007.

[18] Laya Heidari Darani. 2014. Persuasive style and its realization through transitivity analysis: A SFL perspective. *Procedia-social and behavioral sciences* 158 (2014), 179–186.

[19] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=H1edEyBKDS

[20] 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[21] Cicero dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting Offensive Language on Social Media with Unsupervised Text Style Transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 189–194.

[22] Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar Zaiane. 2019. Augmenting Neural Response Generation with Context-Aware Topical Attention. In *Proceedings of the First Workshop on NLP for Conversational AI*. Association for Computational Linguistics, Florence, Italy, 18–31. https://doi.org/10.18653/v1/W19-4103

[23] Nils Erik Enkvist. 2016. *Linguistic stylistics*. Vol. 5. Walter de Gruyter GmbH & Co KG.

[24] Xiaocheng Feng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. 2018. Topic-to-Essay Generation with Neural Networks.. In *IJCAI*. 4078–4084.

[25] Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 1–30.

[26] Yao Fu, Yansong Feng, and John P Cunningham. 2019. Paraphrase Generation with Latent Bag of Words. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., 13645–13656. https://proceedings.neurips.cc/paper/2019/file/5e2b66750529d8ae895ad2591118466f-Paper.pdf

[27] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[28] Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61 (2018), 65–170.

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

[30] A. Gidiotis and G. Tsoumakas. 2020. A Divide-and-Conquer Approach to the Summarization of Long Documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), 3029–3040. https://doi.org/10.1109/TASLP.2020.3037401

[31] Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. 2019. Reinforcement Learning Based Text Style Transfer without Parallel Training Corpus. 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)*. 3168–3180.

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[33] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 2672–2680. http://papers.nips.cc/paper/5423-generative-adversarial-nets

[34] MAK Halliday. 1981. Linguistic function and literary style: An inquiry into the language of William Golding's The Inheritors. *Essays in Modern Stylistics* (1981), 325–60.

[35] Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A Probabilistic Formulation of Unsupervised Text Style Transfer. In *International Conference on Learning Representations (ICLR)*.

[36] 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*. 507–517.

[37] Dirk Hovy and Shannon L Spruit. 2016. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 591–598.

[38] Eduard Hovy. 1987. Generating natural language under pragmatic constraints. *Journal of Pragmatics* 11, 6 (1987), 689–719.

[39] Eduard Hovy and Chin-Yew Lin. 1998. Automated Text Summarization and the Summarist System. In *TIPSTER TEXT PROGRAM PHASE III: Proceedings of a Workshop held at Baltimore, Maryland, October 13-15, 1998*. Association for Computational Linguistics, Baltimore, Maryland, USA, 197–214. https://doi.org/10.3115/1119089.1119121

[40] Zhiqiang Hu, Roy Ka-Wei Lee, Lei Wang, Ee-peng Lim, and Bo Dai. 2020. DeepStyle: User Style Embedding for Authorship Attribution of Short Texts. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*. Springer, 221–229.

[41] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1587–1596.

[42] Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P Xing. 2018. On Unifying Deep Generative Models. In *International Conference on Learning Representations*.

[43] Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. Visual Storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 1233–1239. https://doi.org/10.18653/v1/N16-1147

[44] Farkhund Iqbal, Hamad Binsalleeh, Benjamin CM Fung, and Mourad Debbabi. 2010. Mining writeprints from anonymous e-mails for forensic investigation. *digital investigation* 7, 1-2 (2010), 56–64.

[45] Roz Ivanič. 2004. Discourses of writing and learning to write. *Language and education* 18, 3 (2004), 220–245.

[46] Parag Jain, Abhijit Mishra, Amar Prakash Azad, and Karthik Sankaranarayanan. 2019. Unsupervised controllable text formalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6554–6561.

[47] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rkE3y85ee

[48] Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing Modern Language Using Copy-Enriched Sequence to Sequence Models. In *Proceedings of the Workshop on Stylistic Variation*. 10–19.

[49] Di Jin, Zhijing Jin, Joey Tianyi Zhou, Lisa Orii, and Peter Szolovits. 2020. Hooks in the Headline: Learning to Generate Headlines with Controlled Styles. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[50] Zhijing Jin, Di Jin, Jonas Mueller, Nicholas Matthews, and Enrico Santus. 2019. IMaT: Unsupervised Text Attribute Transfer via Iterative Matching and Translation. 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)*. 3088–3100.

[51] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. 2019. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics* (2019).

[52] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled Representation Learning for Non-Parallel Text Style Transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 424–434.

[53] Barbara Johnstone. 1989. Linguistic strategies and cultural styles for persuasive discourse. (1989).

[54] Barbara Johnstone. 2009. Stance, style, and the linguistic individual. *Stance: sociolinguistic perspectives* (2009), 29–52.

[55] Tomoyuki Kajiwara and Mamoru Komachi. 2016. Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 1147–1158.

[56] Maurits Kaptein, Boris De Ruyter, Panos Markopoulos, and Emile Aarts. 2012. Adaptive persuasive systems: a study of tailored persuasive text messages to reduce snacking. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2, 2 (2012), 1–25.

[57] Maurits Kaptein, Panos Markopoulos, Boris De Ruyter, and Emile Aarts. 2015. Personalizing persuasive technologies: Explicit and implicit personalization using persuasion profiles. *International Journal of Human-Computer Studies* 77 (2015), 38–51.

[58] Soomin Kim, Joonhwan Lee, and Gahgene Gweon. 2019. Comparing data from chatbot and web surveys: Effects of platform and conversational style on survey response quality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[59] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[60] André Klahold and Madjid Fathi. 2020. *Computer Aided Writing*. Springer.

[61] André Klahold and Madjid Fathi. 2020. Word Processing as Writing Support. In *Computer Aided Writing*. Springer, 21–29.

[62] R. Kneser and H. Ney. 1995. Improved backing-off for M-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1. 181–184 vol.1.

[63] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*. 177–180.

[64] Xiang Kong, Bohan Li, Graham Neubig, Eduard Hovy, and Yiming Yang. 2019. An adversarial approach to high-quality, sentiment-controlled neural dialogue generation. *arXiv preprint arXiv:1901.07129* (2019).

[65] Chih-Te Lai, Yi-Te Hong, Hong-You Chen, Chi-Jen Lu, and Shou-De Lin. 2019. Multiple Text Style Transfer by using Word-level Conditional Generative Adversarial Network with Two-Phase Training. 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)*. 3570–3575.

[66] Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. *Advances in neural information processing systems* 29 (2016), 4601–4609.

[67] Guillaume Lample, Sandeep Subramanian, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-Attribute Text Rewriting. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

[68] Wouter Leeftink and Gerasimos Spanakis. 2019. Towards Controlled Transformation of Sentiment in Sentences. In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence, ICAART 2019, Volume 2, Prague, Czech Republic, February 19-21, 2019*. 809–816.

[69] Jochen L Leidner and Vassilis Plachouras. 2017. Ethical by design: Ethics best practices for natural language processing. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*. 30–40.

[70] Dianqi Li, Yizhe Zhang, Zhe Gan, Yu Cheng, Chris Brockett, Bill Dolan, and Ming-Ting Sun. 2019. Domain Adaptive 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)*. 3295–3304.

[71] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A Persona-Based Neural Conversation Model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 994–1003. https://doi.org/10.18653/v1/P16-1094

[72] 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)*. 1865–1874.

[73] Yi Liao, Lidong Bing, Piji Li, Shuming Shi, Wai Lam, and Tong Zhang. 2018. Quase: Sequence editing under quantifiable guidance. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3855–3864.

[74] Dayiheng Liu, Jie Fu, Yidan Zhang, Chris Pal, and Jiancheng Lv. 2020. Revision in Continuous Space: Fine-Grained Control of Text Style Transfer. (2020).

[75] Qian Liu, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. 2020. You Impress Me: Dialogue Generation via Mutual Persona Perception. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 1417–1427. https://doi.org/10.18653/v1/2020.acl-main.131

[76] Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*. 5103–5113.

[77] Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao, and Michel Galley. 2017. Multi-Task Learning for Speaker-Role Adaptation in Neural Conversation Models. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Asian Federation of Natural Language Processing, Taipei, Taiwan, 605–614. https://www.aclweb.org/anthology/I17-1061

[78] Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 5116–5122.

[79] Shuming Ma, Xu Sun, Wei Li, Sujian Li, Wenjie Li, and Xuancheng Ren. 2018. Query and Output: Generating Words by Querying Distributed Word Representations for Paraphrase Generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 196–206. https://doi.org/10.18653/v1/N18-1018

[80] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics, 142–150.

[81] Charles A MacArthur. 2009. Reflections on research on writing and technology for struggling writers. *Learning Disabilities Research & Practice* 24, 2 (2009), 93–103.

[82] David D. McDonald and James D. Pustejovsky. 1985. A Computational Theory of Prose Style for Natural Language Generation. In *Second Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Geneva, Switzerland. https://www.aclweb.org/anthology/E85-1027

[83] Yishu Miao and Phil Blunsom. 2016. Language as a Latent Variable: Discrete Generative Models for Sentence Compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 319–328.

[84] Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating Style Transfer for Text. 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)*. 495–504.

[85] Masahiro Mizukami, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Linguistic individuality transformation for spoken language. In *Natural Language Dialog Systems and Intelligent Assistants*. Springer, 129–143.

[86] Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). 2014. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL.

[87] Ian Muehlenhaus. 2012. If looks could kill: The impact of different rhetorical styles on persuasive geocommunication. *The Cartographic Journal* 49, 4 (2012), 361–375.

[88] Jonas Mueller, David Gifford, and Tommi Jaakkola. 2017. Sequence to better sequence: continuous revision of combinatorial structures. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 2536–2544.

[89] Nikola I. Nikolov and Richard H. R. Hahnloser. 2018. Large-scale Hierarchical Alignment for Author Style Transfer. *CoRR* abs/1810.08237 (2018). arXiv:1810.08237 http://arxiv.org/abs/1810.08237

[90] Kyosuke Nishida, Itsumi Saito, Kosuke Nishida, Kazutoshi Shinoda, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2019. Multi-style Generative Reading Comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2273–2284.

[91] Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 85–91.

[92] Richard Yuanzhe Pang. 2019. The Daunting Task of Real-World Textual Style Transfer Auto-Evaluation. *CoRR* abs/1910.03747 (2019). arXiv:1910.03747 http://arxiv.org/abs/1910.03747

[93] Richard Yuanzhe Pang. 2019. Towards Actual (Not Operational) Textual Style Transfer Auto-Evaluation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*. 444–445.

[94] Richard Yuanzhe Pang and Kevin Gimpel. 2019. Unsupervised Evaluation Metrics and Learning Criteria for Non-Parallel Textual Transfer. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*. 138–147.

[95] 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 on association for computational linguistics*. Association for Computational Linguistics, 311–318.

[96] Sunghyun Park, Seung-won Hwang, Fuxiang Chen, Jaegul Choo, Jung-Woo Ha, Sunghun Kim, and Jinyeong Yim. 2019. Paraphrase Diversification Using Counterfactual Debiasing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6883–6891.

[97] G Parra et al. 2019. Automated Writing Evaluation Tools in the Improvement of the Writing Skill. *International Journal of Instruction* 12, 2 (2019), 209–226.

[98] 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)*. 866–876.

[99] Ella Rabinovich, Raj Nath Patel, Shachar Mirkin, Lucia Specia, and Shuly Wintner. 2017. Personalized Machine Translation: Preserving Original Author Traits. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, Valencia, Spain, 1074–1084. https://www.aclweb.org/anthology/E17-1101

[100] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444* (2017).

[101] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[102] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.

[103] 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)*. 129–140.

[104] Sravana Reddy and Kevin Knight. 2016. Obfuscating gender in social media writing. In *Proceedings of the First Workshop on NLP and Computational Social Science*. 17–26.

[105] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 1998. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 59–66.

[106] Horacio Saggion. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies* 10, 1 (2017), 1–137.

[107] Semiu Salawu, Yulan He, and Joanna Lumsden. 2017. Approaches to automated detection of cyberbullying: A survey. *IEEE Transactions on Affective Computing* 11, 1 (2017), 3–24.

[108] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 86–96.

[109] Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[110] Iulian Vlad Serban, Tim Klinger, Gerald Tesauro, Kartik Talamadupula, Bowen Zhou, Yoshua Bengio, and Aaron C. Courville. 2017. Multiresolution Recurrent Neural Networks: An Application to Dialogue Response Generation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, Satinder P. Singh and Shaul Markovitch (Eds.). AAAI Press, 3288–3294. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14571

[111] Mingyue Shang, Piji Li, Zhenxin Fu, Lidong Bing, Dongyan Zhao, Shuming Shi, and Rui Yan. 2019. Semi-supervised Text Style Transfer: Cross Projection in Latent Space. 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)*. 4939–4948.

[112] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*. 6830–6841.

[113] Tianxiao Shen, Jonas Mueller, Regina Barzilay, and Tommi Jaakkola. 2020. Educating text autoencoders: Latent representation guidance via denoising. In *International Conference on Machine Learning*. PMLR, 8719–8729.

[114] Priscilla Chantal Duarte Silva, Ricardo Luiz Perez Teixeira, and Victoria Olivia Araujo Vilas Boas. 2019. Computational Linguistics: Analysis of The Functional Use of Microsoft Text Word Processor Text Corrector. *International Journal of Linguistics, Literature and Culture, LLC* (2019), 23.

[115] Ilana Snyder. 1993. Writing with word processors: a research overview. *Educational Research* 35, 1 (1993), 49–68.

[116] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, 196–205. https://doi.org/10.3115/v1/N15-1020

[117] 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)*. 3260–3270.

[118] Xiao Sun, Jia Li, Xing Wei, Changliang Li, and Jianhua Tao. 2020. Emotional editing constraint conversation content generation based on reinforcement learning. *Inf. Fusion* 56 (2020), 70–80. https://doi.org/10.1016/j.inffus.2019.10.007

[119] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[120] Bakhtiyar Syed, Gaurav Verma, Balaji Vasan Srinivasan, Anandhavelu Natarajan, and Vasudeva Varma. 2020. Adapting language models for non-parallel author-stylized rewriting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9008–9015.

[121] Sho Takase and Naoaki Okazaki. 2019. Positional Encoding to Control Output Sequence Length. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 3999–4004. https://doi.org/10.18653/v1/N19-1401

[122] Youzhi Tian, Zhiting Hu, and Zhou Yu. 2018. Structured Content Preservation for Unsupervised Text Style Transfer. *CoRR* abs/1810.06526 (2018). arXiv:1810.06526 http://arxiv.org/abs/1810.06526

[123] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[124] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research* 11, Dec (2010), 3371–3408.

[125] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in neural information processing systems*. 2692–2700.

[126] Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-Aware Neural Machine Translation Learns Anaphora Resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 1264–1274. https://doi.org/10.18653/v1/P18-1117

[127] Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable Unsupervised Text Attribute Transfer via Editing Entangled Latent Representation. In *Advances in Neural Information Processing Systems*. 11034–11044.

[128] Li Wang, Junlin Yao, Yunzhe Tao, Li Zhong, Wei Liu, and Qiang Du. 2018. A Reinforced Topic-Aware Convolutional Sequence-to-Sequence Model for Abstractive Text Summarization. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 4453–4460. https://doi.org/10.24963/ijcai.2018/619

[129] Xin Wang, Wenhu Chen, Yuan-Fang Wang, and William Yang Wang. 2018. No Metrics Are Perfect: Adversarial Reward Learning for Visual Storytelling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 899–909. https://doi.org/10.18653/v1/P18-1083

[130] 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)*. 3564–3569.

[131] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.

[132] Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. 2019. A Hierarchical Reinforced Sequence Operation Method for Unsupervised Text Style Transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4873–4883.

[133] Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Mask and infill: applying masked language model to sentiment transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 5271–5277.

[134] Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[135] Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. 2018. Hierarchical recurrent attention network for response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[136] Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. 2017. A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3506–3510.

[137] 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)*. 979–988.

[138] Peng Xu, Yanshuai Cao, and Jackie Chi Kit Cheung. 2019. On Variational Learning of Controllable Representations for Text without Supervision. *CoRR* abs/1905.11975 (2019). arXiv:1905.11975 http://arxiv.org/abs/1905.11975

[139] Peng Xu, Jackie Chi Kit Cheung, and Yanshuai Cao. 2020. On variational learning of controllable representations for text without supervision. In *International Conference on Machine Learning*. PMLR, 10534–10543.

[140] Ruochen Xu, Tao Ge, and Furu Wei. 2019. Formality Style Transfer with Hybrid Textual Annotations. *CoRR* abs/1903.06353 (2019). arXiv:1903.06353 http://arxiv.org/abs/1903.06353

[141] Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*. 2899–2914.

[142] Min Yang, Qiang Qu, Kai Lei, Jia Zhu, Zhou Zhao, Xiaojun Chen, and Joshua Z Huang. 2018. Investigating deep reinforcement learning techniques in personalized dialogue generation. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 630–638.

[143] Min Yang, Zhou Zhao, Wei Zhao, Xiaojun Chen, Jia Zhu, Lianqiang Zhou, and Zigang Cao. 2017. Personalized response generation via domain adaptation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1021–1024.

[144] Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*. 7287–7298.

[145] Di Yin, Shujian Huang, Xin-Yu Dai, and Jiajun Chen. 2019. Utilizing non-parallel text for style transfer by making partial comparisons. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 5379–5386.

[146] Matt Young. 2002. *The technical writer's handbook: writing with style and clarity*. University Science Books.

[147] Ye Zhang, Nan Ding, and Radu Soricut. 2018. SHAPED: Shared-Private Encoder-Decoder for Text Style Adaptation. In *Proceedings of NAACL-HLT*. 1528–1538.

[148] Yi Zhang, Tao Ge, and Xu Sun. 2020. Parallel Data Augmentation for Formality Style Transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[149] Yi Zhang, Jingjing Xu, Pengcheng Yang, and Xu Sun. 2018. Learning Sentiment Memories for Sentiment Modification without Parallel Data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 1103–1108.

[150] Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018. Style Transfer as Unsupervised Machine Translation. *CoRR* abs/1808.07894 (2018). arXiv:1808.07894 http://arxiv.org/abs/1808.07894

[151] Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M Rush, and Yann LeCun. 2018. Adversarially regularized autoencoders. In *35th International Conference on Machine Learning, ICML 2018*. International Machine Learning Society (IMLS), 9405–9420.

[152] Yanpeng Zhao, Wei Bi, Deng Cai, Xiaojiang Liu, Kewei Tu, and Shuming Shi. 2018. Language Style Transfer from Sentences with Arbitrary Unknown Styles. *CoRR* abs/1808.04071 (2018). arXiv:1808.04071 http://arxiv.org/abs/1808.04071

[153] Chulun Zhou, Liangyu Chen, Jiachen Liu, Xinyan Xiao, Jinsong Su, Sheng Guo, and Hua Wu. 2020. Exploring Contextual Word-level Style Relevance for Unsupervised Style Transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[154] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[155] Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics* 46, 1 (2020), 53–93.

[156] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1097–1100.

11 APPENDIX

11.1 Generated Samples

We picked 4 test samples including positive to negative, negative to positive, formal to informal, and informal to formal to show the quality of generated sentences from different models intuitively.

Table 12. Example outputs Yelp datasets.

Model	From negative to positive (Yelp)	From positive to negative (Yelp)
Source	Food was cold and lacking of flavors .	Excellent service , ood food , and nice atmosphere .
DeleteOnly	They were great , with a wide variety of flavors .	I would give _num_ stars if i could .
Template	Fantastic of flavors and fabulous .	Excellent service , good food , and nice atmosphere .
Del&Retri	Of a huge fan of flavors .	A waste of time and money !
B-GST	Food was bland and of no flavors .	Great service , great food , and great atmosphere .
G-GST	The food was cold and lacking of flavor .	The service , nice food , and atmosphere .
PTO	Food was fresh and delicious of flavors .	Horrible service , horrible food , and horrible atmosphere .
UST	-	Slow service , good food , and nice atmosphere .
SMAE	-	Poor service , poor food , and poor atmosphere .
DRLST	Food was delicious and fresh	Service was ok but the food was not bad
BST	Food is delicious , and the staff .	Very poor , and no service and nothing .
CAAE	Food was fresh and good .	However , food was good , and poor , poor atmosphere .
ARAE	Food was cold and lacking .	Excellent service , good food , nice atmosphere and nice atmosphere .
Ctrl-Gen	Food was nicely and engaging of flavors .	Terrible service , not food , and 80s atmosphere .
Multi-Dec	Food was prepared , vegetarian and service .	Disgusting waiter , both came and dry selection at !
Style-Emb	Food was cold and authentic which has .	Excellent service , good food , and nice atmosphere .
DualRL	Food was fresh and great of flavors .	Terrible service , bad food , and rude atmosphere .
DAST	Food was great and clean of flavors .	Poor service , horrible food , and bland failure .
DAST-C	Food was delicious and selection of flavors .	Terrible service , bad food , and tasteless atmosphere .
PFST	Food was fresh and exceptional of flavors .	Horrible service , bad food , and poor service .

Table 13. Example outputs GYAFC datasets.

	From informal to formal (GYAFC)	From formal to informal (GYAFC)
Source	I can onli say...women r complicated...	I would estimate approximately three or four .
DeleteOnly	I can you say ... women r you ...	I would you you three or four .
Template	In my opinion , onli say ... women to engage in complicated ...	I would estimate jus three or four .
Del&Retri	I can you say women r you as well .	I would you you each other four .
B-GST	I can onli say . . .	I would estimate at least three .
G-GST	I am not complicated . . .	I would estimate approximately three or four .
UST	I don't think it is wrong.	I am a girl.
SMAE	I can do anything at all.	I would not do it for you.
DRLST	I am not attracted to women	I would've been in a long hair
BST	If it is not a complicated ...	If i have him and ou
CAAE	I would have	I would dont cheat out with her
ARAE	I think that is a <unk>	I am not sure that i am not.
Ctrl-Gen	I can secretly recall remain excited.	I would shouldnt quit three or guys!
Multi-Dec	I can say ... women r ...	I was me of guy .
Style-Emb	I can , i don ' t very !	I would or two of him .
DualRL	I can <unk> say women women complicated .	I would <unk> thrilled ...
DAST	I can <unk> <unk> family <unk>	I would didn't in super or bs
DAST-C	I can equally equally provide equally	I would luv 1 :) or honey
PFST	I can <unk>	I would <unk> 6 or 6 <unk>