Bacheleor thesis

Evaluation for automatic text style transfer

Jiali Mao, s3552985, j.mao.2@student.rug.nl

Supervisors: Fatih Turkmen & Malvina Nissim & Huiyuan Lai

University of Groningen

November 19, 2021

Abstract

While changing the style of a text automatically, text style transfer tasks require preserving style-irrelevant contents, which is challenging. Automatic evaluation of text transfer systems uses various metrics, but it is not practical due to the absence of a single gold standard. For purposes of assessing the performance of several text style systems and choosing suitable metrics when human evaluation is not available by checking correlations, an expert evaluation on the output generated by several text style transfer systems is conducted.

Acknowledgements

I would like to express my gratitude to Malvina and Fatih for giving me the opportunity to work on this project. Especially Malvina, for her "voluntary" supervision and consistent support. Furthermore, I would like to thank Huiyuan for the guidance throughout this project and patience in answering all my questions, his support has been very helpful to me. I would also like to acknowledge Antonio for providing useful feedback to me. Finally, I cannot forget to thank Allison, Evan, Lucia, Michael, Nicholas, Oliver, Trevor and Yanqiu for their generous time completing the expert evaluation of the project.

Contents

1 Introduction

Text style transfer is a task in natural language generation, which intends to change the style of text but keep the style-independent content unchanged. Based on different styles, such as formality, polarity, offensive, politeness, there are various specific text style transfer tasks [43, 26, 9, 32]. The development of automatic text style transfer is motivated by personalized intelligent writing assistants and other natural language processing tasks, such as paraphrasing, summarization, and style-specific machine translation, as normally it has high investments for humans to conduct these tasks.

Traditional methods of text style transfer usually require pre-defined templates for expressions of styles, expressions with the same meaning but in different styles, and domain-specific knowledge, which constrains the general application as constructions of the templates are time-consuming and needs to be done whenever a new domain or a new style is given. Trying to address the problems of the traditional approaches, neural network-based text transfer systems have been proposed in recent years. They can be categorized into two kinds in general, supervised systems and unsupervised systems. There are three methods to establish a supervised system, multi-tasking, inference techniques, and data augmentation [16]. Due to the lack of parallel data, unsupervised systems are popular as references are not needed. There are also three approaches, in general, to construct unsupervised systems, disentanglement, prototype editing, and synthetic data construction [16].

Despite the fact that the interest in automatic text style transfer is growing, the lack of standardized evaluation is a problem. Based on a survey of neural-based text style transfer [16], three most commonly used criteria for style transfer quality are transferred style strength, semantic preservation, and fluency. Transferred style strength indicates the amount that fits the target style in the output sentence. Semantic preservation assesses how well the output sentence maintains the style-independent information of the source sentence. Fluency measures the possibility that the output sentence could have been written by a native speaker. Similar criteria are defined in [35]. There are two ways to conduct the evaluation, automatic evaluation, and human evaluation. For automatic evaluation, it is common to use content-based automatic metrics and style classifiers. To assess content preservation, both system output sentences and references are fed to content-based metrics, both source sentences, and human references can be used as references. There are various content-based automatic metrics established on different methodologies, for instance, BLEU [40], METEOR [2] and ROUGE [29] are n-gram matching based, BERTScore [51] is embedding based, BLEURT [45] and COMET [44] are neural-based. To assess style strength, one can feed system output sentences to style classifiers to get the amount that fits the target style. Even though automatic evaluation is reproducible and easy to accomplish, it is not flexible, and not all of the automatic metrics achieve promising results compared to human judgments at segment-level (sentence-level) [31]. Human evaluation is more flexible compared to automatic evaluation, one reason is that it can either provide an overall score or separate scores for each criterion. Rating and ranking are two methods frequently used. Different annotators (domain experts and crowd workers) are suggested to use [47]. It can be seen that human evaluation tends to be expensive as the need of domain experts and it is also challenging to be reproduced.

In this project, one particular text style transfer task is performed in the experiment, namely formality transfer. It is desired to change the formality of a sentence but preserve its meaning. For instance, given a formal sentence "How are you doing?", one possible transformation can be "what's up, dude?", which expresses the same meaning but informally. There is a similar task, polarity swap, which aims to change the polarity of a sentence but keep the theme. If "The food is bland here." is provided, then one suitable output for polarity swap would be "The food is delicious here.". Although the theme for these two sentences, remarks on food, is preserved, the meanings are changed, which is different from formality transfer.

We mainly focus on which automatic metrics best correlate with human judgment in the task of formality transfer. Three questions are investigated:

RQ1: Which aspects of human assessment (content preservation, style strength, fluency) correlate best with which metrics?

RQ2: What is a feasible, implementable setting for human evaluation of generated text, also with respect

to observations in the current literature?

RQ3: According to human assessment, which of the existing text style transfer system works best?

According to the research questions, we first set up a human (expert) evaluation for formality transfer using outputs generated by various style transfer systems and corresponding reference sentences. Second, different automatic metrics are employed on the same sentences evaluated by human annotators. Third, both segment-level correlations and system-level correlations are shown between human evaluation and automatic evaluation.

We organize the remainder of the thesis as follows: common evaluation approaches for natural language generation and translation, categorization and introduction of neural text style transfer systems are introduced in section 2. Task definition, the dataset, automatic metrics used, methodology of human evaluation and correlation are presented in section 3. Results and answers to the research questions are shown in section 4. Section 5 concludes this thesis.

2 Related Work

In this section, common evaluation methods of both automatic evaluation and human evaluation are studied in 2.1, several text style transfer systems and 2 models are introduced in 2.2.

2.1 Evaluation Approaches

2.1.1 Human evaluation

Two common human evaluation methods are direct assessments and relative assessments, both of them can evaluate either the overall quality or some quality criteria of generated sentences. For direct assessments, system output sentences are shown to the annotators one at a time, and either the overall quality or some quality criteria of generated sentences are judged compared to the corresponding source sentence or reference sentence by, for example, rating. The scales for direct assessments can be discrete [21, 7, 10, 38] or continuous (specific numerical labels are not shown to the annotators) [33, 13]. Relative assessments are used to compare system output sentences against, for example, human references, system variants and/or baselines. It can be done by having annotators rank a set of sentences, choose some sentences they prefer, choose the best and worst candidates or other ways to compare candidate sentences [20, 18, 36]. To address the problem that relative assessments only provide information about the relative quality of systems compared, magnitude estimation is combined with ranking in RankME [39]. According to Howcroft et al. [14], various quality criterion names are used in the human evaluation of Natural Language Generation and definitions of criterion names are not always provided among these researches, but after normalizing different quality criterion names by using given definitions or relevant information, it is found that usefulness for task/information need, grammaticality, quality of outputs, understandability, and correctness of outputs relative to input (content) are the most frequently used criteria. It also indicates a problem of human evaluation of this area, there is no common standard of which criteria to use, in fact, there is no standardized procedure in general. For text style transfer, it is confirmed from a recent study that the lack of standardization and specification of human evaluation protocols are current problems as well [3]. One solution to this problem is to use evaluation platforms, such as GEM [12]. For annotators, experts and crowd workers are two commonly used types in research in which backgrounds of annotators are specified. For data selection of human evaluation in text style transfer tasks, it is shown that data are usually randomly selected and the regular size of instances evaluated per system is 100 [3].

2.1.2 Automatic evaluation

Both overall quality and separate quality criteria of text style transfer systems can be assessed automatically. Moreover, the most frequent criteria are content preservation, style strength and fluency. It is prevalent to calculate the mean of style strength and the BLEU [40] score between system output sentences and human-written references as the overall score of a text style transfer system [24, 30]. However, BLEU

does not correlate very well with human evaluation, and not all existing style transfer datasets provide references. Another challenge is the absence of a single gold standard for system outputs. To evaluate content preservation, different kinds of automatic metrics can be applied to system output sentences and related source sentences/human references. BLEU is the popular one used [30, 26, 52], while other metrics such as BLEURT [45] and COMET [44] are effective [24, 23]. As a common method to evaluate the style strength, a style classifier is trained to judge whether system output sentences fit the target style, and provides the amount/probability of an output sentence fitting the target style. The style strength of the system is calculated as the ratio of the number of output sentences classified as in the target style and the total number of test output sentences [26, 52]. To assess fluency automatically, perplexity can be computed by a pre-trained language model [49, 52]. However, it is suggested there is no crucial correlation between perplexity and human ratings of fluency [35], which leaves the effectiveness of this automatic method open for discussion.

2.2 Style transfer models and systems

2.2.1 Models

Two main pre-trained models that can be applied in style transfer tasks, namely BART [25] and GPT-2 [42], are introduced in this section.

BART It is an autoencoder built with a sequence-to-sequence model for eliminating noises of text. For implementation, it is a sequence-to-sequence model that combines two transformers, with an encoder and a decoder. The input of the bidirectional encoder is the original text depraved with noises. Afterward, tokens of the original text are fed to the left-to-right auto-regressive decoder, with a mask hiding the future tokens that need to be predicted. Moreover, the return states of the bidirectional encoder served as the initial state of the auto-regressive decoder, so the decoder gained information about how to predict. Within the auto-regressive decoder, tokens can be predicted based on past predictions, therefore, the decoder can be easily used for text generation. The probability to generate the original text can then be computed with the auto-regressive decoder. In addition, on the pre-training for the encoder, different kinds of noise approaches can be applied, which leads to one advantage of BART, it does not have limitations on noising schemes, any text corruptions can be used. For illustration, token masking, token deletion, text infilling, sentence permutation and document rotation. BART is made to minimize the cross-entropy between the original text and the output of the auto-regressive decoder. In some text style transfer systems, both encoder and decoder of BART are used as a single decoder.

GPT-2 Generative Pretrained Transformer 2 is a large language model designed to predict the next token given the preceding text. It is built on Transformer [48] architecture and it can be seen as a stack of Transformer decoder (without encoder-decoder self-attention layer) blocks. The number of Transformer decoder blocks is one of the main distinguishing factors between the four different GPT-2 model sizes (small, medium, large and extra large), the other two factors are the number of parameters and model dimension. Pre-layer normalization is applied in each sub-block (masked self-attention block and feed-forward neural network block) of the decoders. Basically, everything is increased compared to GPT [41], in GPT-2, the vocabulary is 50257, context size is 1052 tokens and the batch size is 512. More importantly, it is trained on massive data. It uses the dataset WebText, which has the text of 45 million web links of Reddit with at least 3 Karma (a score in Reddit). GPT-2 provides the possibility to have a promising performance by using WebText, as it contains a huge amount of natural languages in various domains and contexts.

2.2.2 Systems

Several neural-network-based style transfer systems are presented in this section. As mentioned in the introduction, large amounts of parallel data are required to train the style transfer systems for supervised learning, but there is not enough data to use. Researchers in this area address the problem in different ways, either by "creating" more parallel data or doing it in unsupervised ways.

NMT-Combined[43] Due to the size of parallel data is not big enough to train the Neural Machine Translation (NMT) models, additional synthetic sentence pairs need to be created. One suggested way is to get more sentences using the Phrase-based machine translation (PBMT) model given extra source sentences. The other proposed way is to apply back-translation of the PBMT model given some target sentences from Grammarly's Yahoo Answers Formality Corpus (GYAFC).

Bi-Directional Formality Transfer[37] It is a joined neural machine translation model that is able to transfer both from informal to formal and from formal to informal. To train the model, sentence pairs of each transfer direction are jointed and each source sentence is assigned a tag in the beginning to indicate its target style ($\langle F \rangle$ shows the target style is formal and $\langle I \rangle$ is for informal). In addition, Byte-pair encoding is applied to the joint source and target data and their word embeddings are bound. Moreover, three methods are used to improve the model performance. First, to increase the size of training data, the train set of both *Entertainment & Music* and *Family & Relationships* in Grammarly's Yahoo Answers Formality Corpus (GYAFC) [43] are combined and used to train the model. Second, four randomly seeded models trained on the combined dataset are used in ensemble decoding. Third, sentence pairs from French to English translation tasks that have a close domain or topic of GYAFC are augmented to train the model. As the target sentences of translation tasks are in English and they have similar topics and styles as the training data used for formality transfer, it is hypothesized adding these bilingual data would help and it is verified by the evaluation results.

Lai's[24, 23] Although fine-tuning the above pre-trained models achieve decent results on content preservation, adding style classification reward and BLEU score reward improves the style strength and content preservation for formality transfer task [24].

For a satisfactory output sentence, the style is contrasting to the style of the source sentence, so substantial change in style is rewarded. To quantify the method, confidence of the pre-trained style classifier TextCNN [19] is used. Given the sentence with target style y, the confidence of y in style s (formal or informal) is defined as $p(s|y) = softmax(\text{TextCNN}(y, \theta))$, where θ are parameters of the classifier TextCNN. Then the style classification reward is calculated as $R_{style_{target}} = \lambda_{style}(p(s_{target}|y) - p(s_{source}|y))$, where λ_{style} is the weight for style classification reward, s_{target} is the target style and s_{source} is the style of the source sentence. However, for the GPT-2 based model, apart from generating a sentence in the target style, it also produces a sentence $x^{'}$ without style change but possibly with word change. Therefore, the reward to the source sentenced needs to be added as well, and it is computed as $R_{style_{source}} = \lambda_{style}(p(s_{source}|x') - p(s_{target}|x'))$.

Except for style strength, content preservation serves as an essential criterion. The automatic metric BLEU is introduced in section 2.1.1. To boost content preservation for formality transfer, BLEU score reward is proposed [24]. It is defined as $R_{bleu} = \lambda_{bleu}(bleu(y', y) - bleu(y^s, y))$, where λ_{bleu} is the weight for BLUE score reward, y is the reference sentence, y' is the generated sentence in target style getting by maximizing the distribution of model outputs at each time step and y^s is sampled from the distribution of model outputs at each decoding time step as usual.

Policy gradient method is applied for policy optimization, and the two rewards defined above are used for the estimate of advantage in policy gradient, which indicates if the policy action is better than expected. When the BLEU score reward is used, the policy gradient for both BART-based model and GPT-2 based model are the same, $E[R_{bleu} * \nabla_\phi \log(P(y^s|x;\phi))]$, where E is the expectation, ∇_ϕ is the gradient with respect to ϕ and *phi* are parameters of the model. When the style classification reward is used, for BART-based model, the policy gradient is defined as $E[R_{style} * \nabla_\phi \log(P(y^s|x;\phi))]$. For GPT-2 based model, the policy gradient is

$$E[R_{style_{source}} * \nabla_\phi \log(P(y^s_{source}|x;\phi))] + E[R_{style_{target}} * \nabla_\phi \log(P(y^s_{target}|x, x';\phi))]$$

Large amounts of parallel data of text style transfer can be used to train the systems to achieve remarkable results in content preservation, but they are not always available as they are task-variant and it is not easy to obtain the natural parallel data for some tasks. It is suggested to use other data for further pre-training [23]. For formality transfer, paraphrase data is used (specifically the dataset PARABANK 2 [15], as there

is a sufficient amount of paraphrase data and formality transfer can be seen as one way of paraphrasing[23]. Apart from using all sentence pairs in PARABANK 2, the ones which have more opposite style are chosen, and forms a subset defined as $D_{subset} = \{(a, b) \mid (p(s_1|a)+p(s_2|b))/2 > \sigma\}$, where (a, b) is a sentence pair and σ is the threshold. For polarity transfer, as the meaning is changed, it cannot be seen as paraphrasing, hence it would not be appropriate to use paraphrase data. To solve this problem, synthetic pairs are introduced. To generate synthetic pairs, first, the sentiment scores of words in Sentiwordnet [1] is used to get the polarity of a word in a sentence. Second, sentences containing only one polarity word from YELP [27] are chosen as source sentences. Third, the polarity word is replaced by its antonym in Wordnet [34] and the new sentence is served as an output sentence.

In the training phase, iterative back-translation is applied after further pre-training. There are two models, each of which corresponds to one transfer direction. Although the data is non-parallel in the beginning, the input and output of one model can be used to supervise the training of the other model. To generate good results in content preservation and style strength for the models, corresponding rewards similar to the ones in the supervised system in [24] is used. There are two kinds of rewards as well, one for style strength and one for content preservation. Since only BART is used in this unsupervised system, the reward of style strength is $R_{style} = \lambda_{style}(p(s_{target}|y') - p(s_{source}|y'))$. For the supervised system mentioned above, BLEU is used for content preservation reward, the same is applied $R_{bleu} = \lambda_{bleu}(BLEU(y^{'}_{s_i}, x) - BLEU(y^{s}_{s_i}, x))$ as the source sentence of the opposite model is used to train the model and $y^{s}_{s_i}$ is sampled from the distribution of model outputs at each decoding time step with the target style s_i. Apart from BLEU, a learnable automatic metric BLEURT [45] can be also used in content preservation reward, $R_{bleurt} = \lambda_{bleurt}(BLEURT(y^{s}_{s_i}, x))$. The policy gradient method is again used for policy optimization, and policy gradients are the same as defined in the supervised system in [24]. Then the model using iterative-back translation and all the rewards with the data set D_{subset} achieves the best result for formality transfer.

Lastly, the best model is used to generate synthetic pairs which can be used to supervise the training of the original BART model with all rewards. The source sentences are randomly chosen in GYAFC (for formality transfer) and YELP (for polarity swap), which constitute synthetic pairs with the output sentences generated by the best model. Then the synthetic pairs are used to fine-tune the original BART with all rewards. The new learned model has the best performance for the polarity swap task.

DualRL[30] DualRL is a dual reinforcement learning system that performs text style transfer employing two sequence-to-sequence mapping models, rewards, annealing pseudo teacher-forcing and synthetic sentence pairs. A forward model f with parameter θ and a backward model g with parameter ϕ are pre-trained using synthetic sentence pairs output by a template-based baseline [26]. In addition, two rewards are utilized. One reward is for changing style and a pre-trained binary style classifier [19] is used to evaluate the style strength. Given a source sentence x, a candidate sentence (system output) y and the target style s, the reward for changing style is $R_{style} = P(s|y; \varphi)$ where φ is the training parameters of the style classifier. For content preservation, we would like to see if the candidate sentence preserves the meaning of the source sentence, so the reward of content preservation is $R_{content} = P(x|y; \phi)$ where phi is the training parameter of the backward model. Then the two rewards are combined to get an overall reward using harmonic mean, and it is $R = (1 + \beta^2)\frac{R_{content} * R_{style}}{\beta^2 * R_{content} + R_{style}}$ where β is a weight. For policy optimization, the policy gradient method is applied and the overall reward is used to estimate the advantage in policy gradient. The expectation of the overall reward $E[R]$ would be maximized and its gradient is

$$\nabla_\theta E[R] = \sum_k R_k \nabla_\theta \log(P(y_k|x; \theta))P(y_k|x; \theta) \tag{1}$$

where y_k is the kth generated sentence (output of model f) and R_k is the overall reward of y_k.

Moreover, Annealing pseudo teacher-forcing is applied in DualRL. In the beginning, the initial iteration interval p_0 is set. For each iteration i, firstly the model f_θ is trained. θ is updated using equation (1), if the current iteration is a new start of the iteration interval, next generate a synthetic sentence pair using the models f_θ and g_ϕ updated by the last iteration $i - 1$, then use the synthetic sentence pair to train f_θ by maximum likelihood estimation. It follows to train g_ϕ in a similar fashion. Lastly to update the iteration

interval $p = \min(p_0 * r^{\frac{i}{d}}, p_{\max})$ where r is the increase rate, d is the increase gap, p_{max} is the maximum iteration interval, and then go the next iteration and repeat the whole process.

StyIns[50] Given the source style s_i, target style s_j, Φ_N^i(a set of N sentences with style s_i), Φ_N^j (a set of N sentences with style s_j), source sentence x with style s_i and $x \notin \Phi_N^i$, the generator G would output a sentence y with style s_j. G is composed of three parts, a style encoder, a source encoder and a decoder with attention mechanism. The style instances with target style Φ_N^j are input into the style encoder to model a style space $p(z|\Phi_N^j)$, where z is a learned variable to express a style. For a source encoder, the input is a source sentence and the output is a sequence of corresponding hidden states. The hidden states and z are then given to the decoder.

Three kinds of loss are computed and optimized to train the generator G. First, the reconstruction loss $L_{recon} = -\log P_G(x|x, \Phi_N^i)$, is the loss for the generator G to reconstruct the source sentence given the corresponding style instances. Second, the cycle consistency loss, $L_{cycle} = -\log P_G(x|y, \Phi_N^i)$, where y is the output of generator G given style instances Φ_N^j. Third, the adversarial style loss $L_{style} = -\log P_C(j|y)$, where C is a style classifier. Furthermore, the style classifier C is trained by optimizing the loss $L_C = -[logp_C(i|x) + \log p_C(i|x') + \log p_C(M+1|y)]$, where x' is the output of the generator G given x, Φ_N^i and $M+1$ is a class of generated fake. If a reference sentence is available, then the supervised loss can be employed to train the generator G as well, $L_{super} = -\alpha * E_{q(z|y',\Phi_N^j)}[\log p(y'|z,x) + \log p(z|\Phi_N^j) - \log q(z|y', \Phi_N^i)] + \beta * E_{q(z|\Phi_N^j)}[-\log p(y'|z,x)]$, where α and β are the scaling parameters set in the learning process.

Zhou's[53] Two stages are established for transferring text style in an unsupervised way. Stage 1 builds an attentional sequence-to-sequence model which is able to re-predict style relevance for words. Stage 2 extends the model built on stage 1 to generate sentences using style relevance.

The basic model of stage 1 is a sequence-to-sequence model having one encoder and one decoder with attention mechanism. Both the encoder and the decoder are forward Gated Recurrent Unit (GRU) networks. A source sentence X with words x_i is fed to the encoder, and it outputs a sequence of hidden states. The last hidden state in the output of encoder is then fed to the decoder, and it predicts probabilities. The objective function of the model is $L = L_{recon} + L_{relev}$. The sentence reconstruction loss is $L_{recon}(\theta) = -\sum_{i=1}^{|\bar{X}|} \log P(x_i|x_{<i}, \bar{X})$, where θ is the parameters of the model and $\bar{X}$ is an variation of X with some words randomly being replaced. The style relevance restoration loss $L_{relev}(\theta, \phi) = \frac{1}{|X|} \sum_{i=1}^{|X|} (\lambda_i - \hat{\lambda}_i)^2$, where ϕ is the parameters used to get $\hat{\lambda}_i$, λ_i is the style relevance of the i-th word in the input sentence, and $\hat{\lambda}_i$ is the style relevance of the i-th output word. $\hat{\lambda}_i = \mathbf{v}_\lambda^T tanh(\mathbf{W}_\lambda \mathbf{h}_{i-1}^d)$, where $\mathbf{h}_{i-1}^d$ is the previous decoder hidden state, $\mathbf{v}_\lambda$ and $\mathbf{W}_\lambda$ form the ϕ in L_{relev}. To get λ_i, layer-wise relevance propagation is used to pre-train a TextCNN style classifier [19]. $\lambda_i = tanh(\mu|r(x_i)|)$, where μ is a scaling factor, $r(x_i)$ is the style relevance score and is calculated as the sum of relevance score of neurons at the 0-th layer.

At stage 2, the decoder of the model on stage 1 is extended with a neural style component to update the decoder hidden state to better generate output words considering their target style relevance. The updated decoder hidden state $\hat{\mathbf{h}}_j^d = \mathbf{h}_j^d + \hat{\lambda}_j * \triangle \mathbf{h}_j^d$, where $\mathbf{h}_j^d$ is the decoder hidden state on stage 1, $\hat{\lambda}_j$ is the style relevance of the j-th output word and $\triangle \mathbf{h}_j^d$ is the modification to $\mathbf{h}_j^d$. $\mathbf{h}_j^d = GRU(\mathbf{e}(y_{j-1}), \hat{\mathbf{h}}_{j-1}^d, \mathbf{c}_j)$, $\hat{\mathbf{h}}_j^d = f(\mathbf{e}(y_{j-1}), \hat{\mathbf{h}}_{j-1}^d, s_t; \varphi)$, where GRU is Gated Recurrent Unit, $\mathbf{e}(y_{j-1})$ is the embedding vector of the previous output word y_{j-1}, $\mathbf{c}_j$ is the context vector and is calculated as the weighted sum of all hidden states of the words in the corresponding input sentence, and f is the multi-layer perception function with parameter φ. The objective function used to fine-tune the extended model is $L_2 = L_{st} + \alpha L_{y\lambda} + \beta L_{cp} + \gamma L_{fm}$, where L_{st} is the style transfer loss, $L_{y\lambda}$ is the style relevance consistence loss, L_{cp} is the content preservation loss, L_{fm} is the fluency loss and α, β, γ are weighting factors. Each loss term would be introduced now in details. The style transfer loss is used to measure how well the output sentence fits the target style, and it is computed as $L_{st}(\theta, \phi, \varphi) = -E_{(X,s)\sim D}(\log p(s_t|G(Y)))$, where X is the source sentence with style s, D is the train set, s_t is the target style and $G(Y)$ is the output sentence according to gumbel-softmax distribution. The style relevance consistency loss shows if the predicted style relevance of the output words

are the same as output of the classifier, $L_{y\lambda}(\theta,\phi) = \frac{1}{|Y|}\sum_{j=1}^{|Y|}(\lambda_j - \hat{\lambda}_j)^2$, where $|Y|$ is the number of words in the sentence Y. The content preservation loss indicates to what extent the output sentence remain the same meaning as the source sentence, $L_{cp}(\theta,\phi,\varphi) = (\sum_i^{|X|}(1-|\lambda_i|)\mathbf{e}(x_i) - \sum_j^{|Y|}(1-|\lambda_j|)\mathbf{e}(y_j))^2$. Lastly, the fluency modelling loss is used to show if the model is able to generate fluent sentences. A bidirectional Gated Recurrent Unit based language model is pre-trained using sentences with target style and then used in calculating the fluency modelling loss. $L_{fm} = (L_{fm\_f} + L_{fm\_b})/2$, where fluency modelling loss in the forward direction $L_{fm\_f}(\theta,\phi,\varphi) = \sum_{j=1}^{|Y|}\mathbf{P}(*|y_{<j},X)^T\log(\mathbf{P}_f(*|y_{<j}))$, and L_{fm_b} is the fluency modelling loss in the back direction and is computed in a similar way.

Style-Transformer[6] It employs the Transformer model [48] on the style transfer tasks. Given a source sentence x, the Transformer encoder outputs a sequence of states $\mathbf{h}$. Then using $\mathbf{h}$ and the given target style s, the Transformer decoder predicts the probability of the output sentence y auto-regressively, $p_\theta(y|x,s) = \prod_{t=1} p_\theta(y_t|z,y_{<t})$, $p_\theta(y_t|z,y_{<t}) = softmax(\mathbf{d}_t)$, where θ is the parameter of the Transformer model and $\mathbf{d}_t$ is the output logit vector of the Transformer decoder at timestep t.

For output sentences of the Transformer model, to get better control of their style, the discriminator network is established. Two discriminators are chosen. First, the conditional discriminator, whose inputs are a sentence x and a style s, and the output is whether the given sentence fits the given style (c=1 indicates yes), the loss function used in training is $-p_\phi(c|x,s)$. Second is multi-class discriminator, which takes a sentence x as the input and it is assigned to a specific class, the corresponding loss function is $-p_\phi(c|x)$. The parameter ϕ is updated by minimizing the loss function.

Style-Transformer is an unsupervised approach, it addresses the problem of lacking reference sentences using self reconstruction, cycle reconstruction and style controlling. For self reconstruction, the input sentence x and its style s are fed into the model and the model tries to reconstruct the input sentence. The reference in this case are simply the input sentence. The loss function $L_{self}(\theta) = -p_\theta(y = x|x,s)$, where y is the output of the model given x and s. For cycle reconstruction, the output sentence $\hat{y}$ of the model given the source sentence x and the style s is fed to the model as an input to rebuild the sentence x. The model is learned to minimize the loss $L_{cycle}(\theta) = -p_\theta(y = x|\hat{y},s)$. For both self reconstruction and cycle reconstruction, the reference sentence is simply the source sentence x. Moreover, for cycle reconstruction, it makes the model to learn content preservation. The style controlling loss is used to maximize the probability of style $\hat{s}$, given $\hat{y}$ to the discriminator. If the conditional discriminator is used, $L_{style}(\theta) = -p_\phi(c = 1|\hat{y},\hat{s})$. If the multi-class discriminator is applied, $L_{style}(\theta) = -p_\theta(c = \hat{s}|\hat{y})$. Then L_{self}, L_{cycle} and L_{style} are added and used to update the parameter θ of the transformer model.

DGST[28] As another unsupervised system, DGST has two transformers (f and g) in order to achieve cycle reconstruction. A sentence x with style s_x is given to f, and f outputs a sentence y' with style s_y. A sentence y with style s_y is fed to g to get a sentence x' with style s_x. It is then practicable with cycle reconstruction, given a source sentence to f to get an output sentence, then feed it to g to obtain a reconstruction of the source sentence. The loss function used to train DGST is $L = L_f^c + L_g^c + L_f^s + L_g^s$, where L_f^c and L_g^c impose the system to preserve content from source sentences, L_f^s and L_g^s enforce a style change. For preserving content, f and g are trained to denoise using neighbourhood sampling. A neighbourhood of a sentence y $N(y,\gamma)$, is a set of sentences containing y and all its noisified sentences with noise intensity γ. γ determines the proportion of words modified in the original sentence. $L_f^c = E_{\hat{y}\sim N(y,\gamma)}D(y||f(\hat{y}))$, where $\hat{y}$ is sampled from $N(y,\gamma)$, $D(y||f(\hat{y}))$ is the Hamming distance or Levenshtein distance between y and $f(\hat{y})$. Similarly, $L_g^c = E_{\hat{x}\sim N(x,\gamma)}D(x||g(\hat{x}))$. For style change, $L_f^s = E_{\hat{x}'\sim N(g(y),\gamma)}D(y||f(\hat{x}'))$ and $L_g^s = E_{\hat{y}'\sim N(f(x),\gamma)}D(x||g(\hat{y}'))$.

DelRetri[27] In this system, the words used as style markers are removed from the source sentence, a sentence with target style markers and similar content of the source sentence is retrieved from a corpus, then the remaining part of the source sentence and the one retrieved are fed into a neural model to generate the final output sentence.

The first step is deleting. S is the set of possible styles, for the sake of deleting style markers, they are

identified first. The salience of a marker m with respect to style $s \in S$, is defined as $salience(m, s) = \frac{\lambda + count(m, D_s)}{\lambda + \sum_{s' \in S, s' \neq s} count(m, D_{s'})}$, where λ is the smoothing parameter, D_s is the set of sentences with style s in the corpus and $count(m, D_s)$ is the number of occurrences of m in D_s. m is a marker if $salience(m, s) > \gamma$, where γ is a threshold. Given the source sentence x with style s_{src}, $marker(x, s_{src})$ is the set of all style markers in x, $content(x, s_{src})$ is the remaining sequence of words after removing all markers in $marker(x, s_{src})$ from x. For the second step, to have more knowledge of what markers of the target style should be added, the sentence x_{tgt} with the target style and having the closest content with the source sentence is retrieved. $x_{tgt} = \arg\min_{x' \in D_{s_{tgt}}} d(content(x, s_{src}), content(x', s_{tgt}))$, where d is a distance metric that can be applied to compare two sentences, such as Euclidean distance. The third step is generating the output sentence. Given $content(x, s_{src})$ and $marker(x_{tgt}, s_{tgt})$ to a recurrent neural network, it generates a sentence y with style s_{tgt} by selecting a place to insert the style markers and adjusting function words to achieve better fluency. As another unsupervised method, for training the neural network, self reconstruction and denoising are applied. The network is trained to minimize the loss $L(\theta) = \sum_{(x, s_{src} \in D)} \log p(x | content(x, s_{src}), marker'(x, s_{src}); \theta)$, where $marker'(x, s_{src})$ is noisified $marker(x, s_{src})$ by randomly changing each marker independently with probability 10% and θ is the parameter of the neural network used.

3 Methods

In this section, first of all, the definition of formality transfer and the dataset used to get source sentences are introduced in 3.1. Content-based metrics and style classifiers used in the experiment are investigated in 3.2. In 3.3, formulations of the expert evaluation are shown in detail. Finally, two correlation methods used are presented in 3.4.

3.1 Task and dataset

Formality transfer Define $S = \{\text{“}formal\text{”}, \text{“}informal\text{”}\}$, given a source sentence with style $s \in S$, the task is to generate a sentence with style $s' \in S$, $s' \neq s$ but preserve the meaning of the source sentence. There are four examples in Table 1.

For formality transfer, Grammarly's Yahoo Answers Formality Corpus (GYAFC) [43] is used. It selects sentences from two domains *Entertainment & Music* and *Family & Relationships* in the question-answering forum, Yahoo Answers, as informal sentences. Then, crowd workers are required to rewrite those informal sentences to formal style (one rewrite for each informal sentence), and the resulting sentence pairs configure the GYAFC train set. For the tune set and test set, human experts are involved. For the informal to formal direction, four rewritten sentences in the formal style written by experts are collected for each informal sentence and served as references. For the formal to informal direction, three rewritten sentences in informal style written by experts are collected for each formal rewrite from the informal to formal direction. Then the three rewritten sentences and the initial informal sentence are composed to be references.

For the evaluation set up in this project, 80 sentences are randomly selected from GYAFC train set with domain *Family & Relationships* as source sentences. Among these sentences, half of them are formal and half are informal.

Style transfer direction	Original sentence	Transformed sentence
informal → formal	it all depends on when ur ready.	It all depends on when you are ready.
informal → formal	The Best of Luck to ya!	I wish you the best of luck.
formal → informal	In my opinion, they do.	I myself think that they do.
formal → informal	You can save money with respect to flowers.	a big savings area is flowers.

Table 1: Examples of formality transfer

3.2 Automatic Evaluation

To evaluate the quality of style transfer systems, it is intuitive that humans can judge the output sentences of systems. However, designing and conducting human evaluation experiments are expensive and can be latent, it is not practical to have a human evaluation as a routine of evaluation for text style transfer in general. Automatic evaluation is thus needed, several metrics are established to automatically compare the output sentences of style transfer systems and the corresponding source sentences or human references, in order to assess the performance of systems for content preservation. Automatic evaluation is cheap and fast compared to human evaluation, as it does not have a time-consuming setup process. This section focuses on 7 content-based metrics and 2 style classifiers used in this project, content-based metrics are categorized in Table 2.

Categorization	Automatic metric
N-gram matching based	BLEU METEOR ROUGE
Embedding based	BERTScore WMD
Learnable	BLEURT COMET

Table 2: Automatic metrics used in this project

3.2.1 Content-based metrics

n-gram matching metrics have been widely applied. The number of consecutive sequences of n words occurs in both output of the transfer system and the reference provided by humans is counted and the fraction is calculated. There are considerable automatic metrics that fall in this category, three of them would be introduced and used in the experiments.

BLEU As one of the first automatic metrics which correlates with human evaluation relatively high, BLEU is applied universally [40]. The foundation of BLEU is the precision measure, which in unigram, is calculated by dividing the number of words in system output that also occur in the reference by the total number of words in system output. In addition, to avoid unreasonable high precision caused by multiple occurrences of the same word in system output that also occurs in a reference, the maximum count in a reference for the word is taken if its count is larger than the maximum count in the reference. BLEU is not only computed for unigram (n = 1), but also for n = 2, 3, 4. In general, each element of an n-gram in the reference can be paired at most once. Moreover, to accommodate test corpora that contain more than one sentence, the number of matched n-gram pairs is summed for each sentence in the corpus and is divided by the number of n-grams for all output sentences of a system. As mentioned before, BLEU is calculated for different n-gram sizes. The precision values of different n-gram sizes are associated using geometric averages. Another point is that BLEU enforces a brevity penalty factor for short output sentences of a system.

METEOR To develop further on BLEU and strengthen some weaknesses of it, METEOR is introduced [2]. It focuses on the lexical level and uses unigrams only. It first restricts the number of matches of each unigram in a sentence such that every unigram of one candidate/reference sentence can match to at most one unigram of the corresponding reference/candidate sentence. Next, it has four methods to match unigrams of a candidate sentence and unigrams of a reference: exact matching of the surface forms, match after stemming and synonyms mapping. However, the latter two methods require external resources such as a stemmer and a synonyms lexicon. In addition, the mapping set with minimum total unigram mapping crosses is chosen. Then it computes the unigram precision (P_1) as the ratio of the number of words in a candidate sentence that matched to words in the reference and the total number of words in the candidate sentence, unigram recall (R_1) as the same numerator of P_1 but denominator being the total number of words in the reference.

Moreover, the harmonic mean of P_1 and R_1 with parameter α, $F_{mean} = \frac{P_1 R_1}{\alpha P_1 * (1-\alpha) R_1}$. $Penalty$ is also defined, as $\beta(\frac{c}{m})^{\gamma}$, where c is the number of consecutive matches that are in the same order as in both candidate and reference, m is the total number of matches, β and γ are parameters. Above all, METEOR is calculated as $F_{mean} * (1 - Penalty)$.

ROUGE It is originally designed as an automatic metric for evaluating summarization of text documents[29]. But it can be used to assess text style transfer tasks as the outputs are also text. Unlike BLEU, the keystone of ROUGE is the recall measure. There are four variations of ROUGE based on diverse modifications of recall. However, only ROUGE-N and ROUGE-L are introduced here as ROUGE-1, ROUGE-2 and ROUGE-L are used in the experiment. ROUGE-N is computed as the number of n-grams in the reference that also occur in the candidate divided by the total number of n-grams in the reference. It is often applied to uni-grams (n = 1) and bi-grams (n = 2). If there are several references for one candidate sentence, the ROUGE-N score for the candidate and each of the references is calculated and the maximum value is set as the score of the candidate. ROUGE-L is the Longest Common Sub-sequence (LCS) based F-measure. If we have a candidate c with length l_c and a reference r with length l_r, then the LCS based precision P is $\frac{LCS(c,r)}{l_c}$, the LCS based recall R is $\frac{LCS(c,r)}{l_r}$ and the LCS based F-measure (ROUGE-L) is $\frac{(1+\beta^2)*P*R}{R+\beta^2*P}$, where β is a positive real factor. In addition, the jackknifing resampling is applied in implementing ROUGE. It sequentially leaves out one reference from the given references and selects the maximum ROUGE score from the candidate and the left references, then takes the average of the maximum scores as the result.

One main pitfall of n-gram based automatic metrics, especially bad for text style transfer tasks, is that it is challenging to evaluate paraphrases sturdy. Owing to the fact that an output sentence of a text style transfer system can be regarded as a poor result since its surface form is distinct from the surface form of the reference, even if the output sentence is semantically correct. Another disadvantage is that n-gram based automatic metrics cannot get the distant dependencies in a sentence, and they do not observe the ordering adjustments that change the content of the sentence.

BERTScore It addresses the two disadvantages of n-gram based automatic metrics mentioned above. It uses contextual embedding to generate the vector representation for each token in a sentence [51]. The advantage of applying contextual embedding is that different vector representations can be provided for the same word in different sentences if the contexts are various. The main model used to generate the contextual embedding is BERT [8]. Moreover, cosine similarity is used to compute the similarity between two tokens. For the sake of clarification, suppose there is a reference sentence x which contains tokens $x_1, x_2, ...$ and a candidate sentence y with tokens $y_1, y_2, ...$ The cosine similarity between two tokens $\mathbf{x_1}$ and $\mathbf{y_1}$ is $\mathbf{x_1}^{\top}\mathbf{y_1}$, where $\mathbf{x_1}$ is the normalized vector representation for a and $\mathbf{y_1}$ is the normalized vector representation for b. To get BERTScore recall, first match each token of the reference sentence to the token in the candidate sentence which achieves the maximum cosine similarity,

$$Recall_{bertscore} = \frac{1}{|x|} \sum_{x_i \in x} \max_{y_j \in y} \mathbf{x_i}^T \mathbf{y_j}$$

BERTScore precision is calculated similarly, but first match each token of the candidate sentence to the token in the reference that would give the maximum cosine similarity,

$$Precision_{bertscore} = \frac{1}{|y|} \sum_{y_j \in y} \max_{x_i \in x} \mathbf{x_i}^T \mathbf{y_j}$$

. On top of that, the inverse document frequency is incorporated to BERTScore as importance weighting and a baseline rescaling is used for readability.

WMD [22] The Word Mover's Distance is introduced as another embedding based automatic approach to measure the dissimilarity between two text documents (two sentences). It is defined as the minimum cumulative cost required to move all words from the normalized bag of word representation of one document $\mathbf{d}$ to the normalized bag of word representation of another document $\mathbf{d}^{'}$, $\min_{\mathbf{T} \geq 0} \sum_{i,j} \mathbf{T}_{ij} c(i,j)$, with two constraints, $\forall i (\sum_j \mathbf{T}_{ij} = d_i)$ and $\forall j (\sum_i \mathbf{T}_{ij} = d_j^{'})$, where $c(i,j)$ is the Euclidean distance between the vector of

word i and the vector of word j in the *word2vec* embedding space, $\mathbf{T}$ is a flow matrix, $\mathbf{T}_{ij} \geq 0$ is how much of word i travels to word j, d_i is the normalized frequency of i in $\mathbf{d}$ and $d_j^{'}$ is the normalized frequency of j in $\mathbf{d}^{'}$.

BLEURT BERT [8] is used in the quality evaluation of BLEURT [45], on top of the regular BERT pre-training, additional pre-training on synthetic data is applied. To anticipate more variations that Natural Language Generation systems may output for better learning of BLEURT, rather than using existing datasets, synthetic sentences pairs are produced by using mask-filling, back translation and randomly dropping out words. Mask-filling is used to get variants in lexical level, firstly two maskings are applied, either mask some tokens randomly or mask tokens that are adjoining sequences, then BERT is used to fill in the masked tokens in the incomplete sentence. Back translation is chosen to get variants in sentence level, as it provides paraphrases. For back translation, an English sentence is translated to another language, then it is translated back to English. The method of randomly dropping out words is used to simulate possible noises or omissions that could be generated by NLG systems. After getting the synthetic sentence pairs, each pair is assigned a set of pre-training signals $\{\tau_t\}$, where τ_t is the target vector of a pre-training task t and there are 9 pre-training tasks. For each pre-training task, the corresponding pre-training signal and loss type are shown in the following:

Pre-training task	Pre-training signals	Loss type
BLEU	$\tau_{\mathbf{BLEU}}$	Regression
ROUGE	$\tau_{\mathbf{ROUGE}} = (\tau_{ROUGE\_precision}, \tau_{ROUGE\_recall}, \tau_{ROUGE\_Fscore})$	Regression
BERTScore	$\tau_{\mathbf{BS}} = (\tau_{BS\_precision}, \tau_{BS\_recall}, \tau_{BS\_Fscore})$	Regression
Backtranslation Likelyhood	$\tau_{\mathbf{en-fr,y|y'}},\ \tau_{\mathbf{en-fr,y'|y}},\ \tau_{\mathbf{en-de,y|y'}},\ \tau_{\mathbf{en-de,y'|y}}$	Regression
Entailment	$\tau_{\mathbf{entail}} = (\tau_{entail}, \tau_{contradict}, \tau_{neutral})$	Classification
Backtranslation flag	$\tau_{\mathbf{bt\_flag}}$	Classification

For BLEU, ROUGE and BERTScore, the pre-training signals are the corresponding automatic metric scores (with precision, recall and f-score used for the ROUGE and BERTScore). For backtranslation likelyhood, French and German are used, and the pre-training signal $\tau_{\mathbf{en-fr,y|y'}} = \frac{\log P(y|y')}{|y|}$, where y is the reference sentence, y' is the candidate sentence / system output sentence, $|y|$ is the number of tokens in y and $P(y|y') = P_{fr \to en}(y|y^*_{fr})$, $y^*_{fr} = \arg\max P_{en \to fr}(y_{fr}|y)$. For entailment, the pre-training signal is composed by three probabilities, entail, contradict and neutral. For backtranslation flag, the signal is a boolean suggesting whether a synthetic sentences is made by applying backtranslation.

Since there are multiple pre-training tasks, an aggregate loss function for pre-trainging is used:

$$l_{pre-training=\frac{1}{E}} \sum_{e=1}^{E} \sum_{k=1}^{K} \gamma_k l_k(\tau_k^m, \hat{\tau}_k^m)$$

where m is a sentence pair, E is the number of sentence pairs, K is the number of pre-traing tasks used, γ_k are hyper-parameter weights obtained with grid search and $l_k(\tau_k^m, \hat{\tau}_k^m)$ is the loss function for pre-training task k. If k is a regression task, $l_k = \frac{||\tau_k - \hat{\tau}_k||^2}{the\ dimension\ of\ \tau_k}$, $\hat{\tau_k} = \mathbf{W}_{\tau_k}\mathbf{v}_{[CLS]} + \mathbf{b}_{\tau_k}$ where $\mathbf{v}_{[CLS]}$ is the vector representation for the special classification token generated by BERT, $\mathbf{W}$ is the weight matrix, and $\mathbf{b}$ is the bias vector.

COMET [44] The pre-trained, cross-lingual encoder XLM-RoBERTa [5] is used as the encoder model. The source sentence, reference sentence and candidate sentence are all feed to the encoder. It generates an embedding for each token in a sentence and each layer. Instead of using only the embeddings output by the last layer of the encoder, the pooling layer is introduced to pool the information of the most significant layer of the encoder to an embedding for each token in a sentence. The embedding of a token x is then $\mu\mathbf{E}_x\alpha$, where μ is a weight coefficient used for training, $\mathbf{E}_x$ is a vector of the embedding of token x in each layer, namely $[\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, ..., \mathbf{e}^{(m)}]$, α is the vector of trainable normalized weight coefficient for each layer (apart from layer 0), $\alpha = softmax([\alpha^{(1)}, \alpha^{(2)}, ..., \alpha^{(m)}])$. It is also suggested to set the weight of one layer to $-\infty$ with a set probability to avoid overfitting. Then the average pooling is applied to the embedding of each token in a sentence to get the sentence embedding. Therefore, the sentence embedding s for the source sentence, the

sentence embedding r for the reference sentence and for the candidate sentence h are obtained.

For the estimator model, after getting the sentence embeddings, two element-wise product, $h \odot s$, $h \odot r$ and two absolute element-wise difference $|h - s|$, $|h - r|$ are calculated. Then the four of them are concatenated to h and r to form a vector, namely $[h, r, h \odot s, h \odot r, |h - s|, |h - r|]$, and it is fed to a forward neural network, which is then trained to minimize the mean square error between predicted scores of candidate sentences and the corresponding reference scores.

For the translation ranking model, apart from a source sentence, a reference sentence, two candidate sentences are also given, where one candidate sentence has a higher ranking than the other. They are first feed to the encoder then after the pooling layer, the corresponding sentence embeddings $\mathbf{s}, \mathbf{r}, \mathbf{h}^+$ and $\mathbf{h}^-$ are generated. Next, the triplet margin loss is used to optimize the embeddings, and it is computed as $\max\{0, d(\mathbf{s}, \mathbf{h}^+) + \epsilon - d(\mathbf{s}, \mathbf{h}^-)\} + \max\{0, d(\mathbf{r}, \mathbf{h}^+) + \epsilon - d(\mathbf{r}, \mathbf{h}^-)\}$, where $d(\mathbf{a}, \mathbf{b})$ is the euclidean distance between a and b, ϵ is the margin. Therefore, it is trained to ensure $|d(\mathbf{s}, \mathbf{h}^+) - d(\mathbf{s}, \mathbf{h}^-)| > \epsilon$ and $|d(\mathbf{r}, \mathbf{h}^+) - d(\mathbf{r}, \mathbf{h}^-)| > \epsilon$.

3.2.2 Style classifiers

TextCNN [19] The essential parts of TextCNN (Text Convolutional Neural Networks) are the convolutional layer and max-over-time pooling layer [4]. Suppose the source sentence has n words, each word is represented by a k-dimensional word vector. Then it has height 1, width n and k input channels. The calculation of TextCNN is first performed by defining multiple one-dimensional convolution kernels and applying the kernels to calculate the convolutions on the input channels to get output channels. Next, max-over-time pooling is applied on the output channels, the output of max-over-time pooling for each output channel is concatenated to a vector, dropout is performed on the resulting vector for regularization. Then the processed vector is fed to a fully connected softmax layer to generate probabilities for each class/category.

fastText [17] A sentence representation is the average of word representations for each word in the sentence. Sentence representations form the hidden variables. Suppose there are N sentences in total, the negative log-likelihood over the classes $-\frac{1}{N}\sum_{n=1}^{N} y_n \log(softmax(\mathbf{W}x_n))$ is minimized, y_n is the class label for the n-th sentence, $\mathbf{W}$ is the weight matrix and x_n is the normalized bag of features of the n-th sentence. It is trained by using stochastic gradient descent and the learning rate used is linearly decaying. To reduce the computational complexity, especially for multiple-classes classifications, hierarchical softmax is applied. In addition, instead of using bag of words as feature representations, bag of n-grams is used for efficiency.

3.3 Human evaluation

Automatic evaluation is not perfect, in order to have more reliable results, human evaluation is also conducted.

Following "Best practices for human evaluation of automatically generated text" summarized in [47], and aspects of evaluation used in [35], instead of having an overall quality assessment, three criteria are chosen: (1) style strength: the amount that the transformed sentence fits the target style; (2) content preservation: the amount of content in the transformed sentence that remains unchanged compared to the corresponding source sentence; (3) fluency: the possibility of the transformed sentence written by a native speaker. The corresponding statements for three criteria shown to the annotators are: (1) The transformed sentence fits the target style; (2) The content of the transformed sentence is the same as the original sentence; (3) Considering the target style, the transformed sentence could have been written by a native speaker.

For measurement, from the results in [47], discrete Likert scales are the popular rating method used in Natural Language Generation (NLG) evaluation (text style transfer can be considered as one kind of NLG task), and 7-point scales have the best performance among all point scales. However, when discrete Likert scales are compared to continuous scales with respect to the reliability of human evaluation in [39], it is shown that continuous scales outperform discrete Likert scales. Therefore, we set up the continuous scale from 1 to 100 (numerical labels are not shown to the annotators), which is also easy to be transformed to other ranges, such as 1 to 7, if needed. For each criteria statement mentioned in the last paragraph, a slider with a set range is shown to the annotators, they could move the sliders to indicate how much they agree or disagree

with the statement, then the corresponding score is collected. An example evaluation task in the interface is shown in Figure 1. Evaluations of two transfer directions are shown alternately. Before showing the actual evaluation tasks to the annotators, a short introduction, specific instructions to conduct the tasks, examples of evaluation for each transfer direction (formal → informal, informal → formal) are provided.

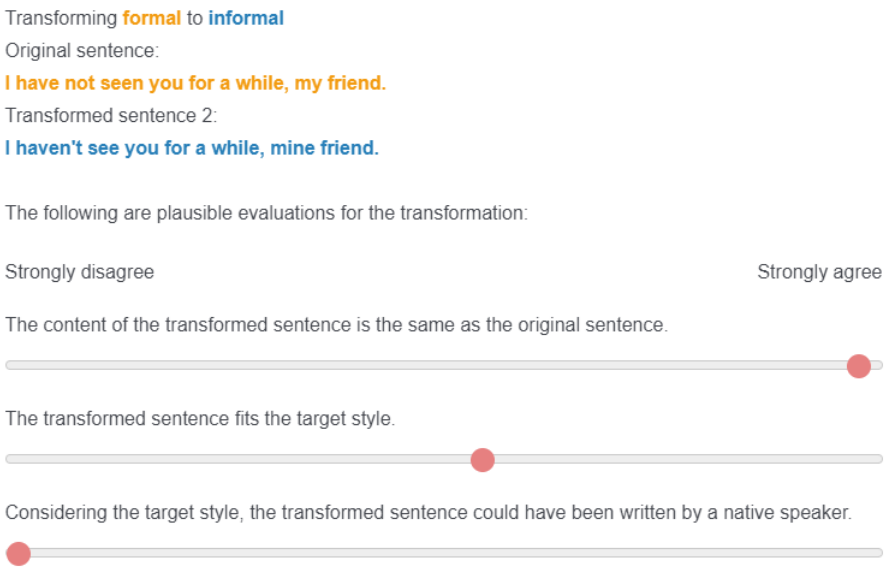

Figure 1: Example of an evaluation

Due to the time constraint of this bachelor project, only expert evaluation is conducted and the crowd worker evaluation for the same data is left for future work. The detailed settings are discussed as follows. For each transfer direction, 40 source sentences for both formal and informal style from GYAFC [43] of domain *Family & Relationships* are randomly chosen, thus there are in total 80 source sentences. 8 text transfer systems from section 2.2.2 are chosen to use in evaluating text formality style, ***NMT-Combined***[43], ***Bi-Directional Formality Transfer***[37], ***Lai's (BART large+SC+BLEU, Trained with High-Quality Generated Paires, IBT+SC+BLEU+BLEURT)***[24, 23], ***DualRL***[30], ***StyIns***[50] and ***Zhou's***[53]. For each source sentence, the output sentence of each system and the corresponding reference sentence are evaluated. The duplicates of output sentences from different systems are eliminated, so there are up to 9 evaluations for one source sentence. Due to a large number of evaluations, the task is divided into 4 sub-tasks, each of them contains transformed sentences of 20 source sentences (10 in each transfer direction). Each sub-task is assigned to two experts so that each sentence is evaluated twice. In total, 8 experts are needed. We have asked 6 English native speakers and 2 non-native English speaker with a linguistics background to perform the tasks.

3.4 Correlation

To compare human evaluation and automatic metrics, correlation analysis needs to be conducted. Two types of correlation analysis are introduced in this section.

Segment-level correlation As a segment in text style transfer tasks (in most cases of NLG tasks) is a sentence, segment-level correlation is the same as sentence-level correlation. Since sentence-level correlation is more relevant for the evaluation of NLG systems [46], to investigate which aspects of human judgment correlate best with which automatic metrics, Kendall's Tau-like formulation (τ) [31] is applied. For both human evaluation and automatic metrics (section 3.3), a score is assigned to a candidate-reference sentence pair for each criterion (mentioned in section 3.4). For two candidate-reference pairs, if the score relationship of human evaluation is consistent with the score relationship of an automatic metric, it is a concordant,

otherwise, it is a discordant. For instance, given two candidate-reference sentence pairs 1 and 2, the score relationship based on human evaluation is $s_1 < s_2$, but for an automatic metric, it is $s_1 > s_2$, it is a discordant. Then the Kendall's Tau-like formulation $\tau = \frac{|concordant|-|discordant|}{|concordant|+|discordant|}$, where $|x|$ is the number of x.

System-level correlation Although segment-level correlation are more significant, it is still essential to use system-level correlation to study the performance of different systems. For a system containing n candidate sentences, the Pearson correlation between human evaluation and a metric M is $r = \frac{\sum_{i=1}^{n}(H_i-\bar{H})(M_i-\bar{M})}{\sqrt{\sum_{i=1}^{n}(H_i-\bar{H})^2}\sqrt{\sum_{i=1}^{n}(M_i-\bar{M})^2}}$, where H_i is the human evaluation score for the i-th candidate-reference pair and $\bar{H} = \frac{\sum_{i=1}^{n} X_i}{n}$.

4 Results

4.1 RQ1

RQ1: Which aspects of human assessment (content preservation, style strength, fluency) correlate best with which metrics?

(All correlation coefficients presented in this section are processed by taking corresponding absolute values as the focus is not whether expert assessments and metrics have positive or negative relations.) In this part, when running content-based metrics, source sentences are first used as "references". In this setting, human references provided by the dataset GYAFC can be seen as outputs of another system since human references are some of the possible variations that would be predictions of a trained system. For content preservation, Pearson correlation coefficients between expert judgments and content-based automatic metrics are calculated for each system and are presented in Table 3. It can be seen not all systems correlate best with the same metric, but after calculating the mean of all systems for each metric, BERTScore has the highest value and BLEURT has the second highest. Moreover, segment-level correlation (Kendall's tau) between output sentences of all systems and each content-based metric are shown in Table 6, for content preservation, BERTScore again correlates best.

	BLEU	METEOR	WMD	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	BLEURT	COMET
BART large + SC + BLEU	0.310	0.223	0.249	0.222	0.222	0.224	**0.358**	0.270	0.182
Trained with High-Quality Generated Pairs	**0.212**	0.100	0.080	0.096	0.098	0.096	**0.212**	0.055	0.114
IBT + SC + BLEU + BLEURT	0.192	0.111	0.016	0.227	0.167	0.219	**0.557**	0.220	0.116
DURL	0.657	0.709	0.706	0.555	0.675	0.564	0.773	**0.792**	0.679
Bi-Directional Formality Transfer	0.204	0.139	**0.229**	0.165	0.195	0.165	0.187	0.142	0.063
NMT-Combined	0.226	0.171	0.199	0.217	0.247	0.200	**0.497**	0.359	0.158
Human references	0.316	0.037	0.318	0.378	0.313	0.363	**0.495**	0.307	0.09
StyIns	0.460	0.415	0.470	0.272	0.300	0.272	0.639	**0.674**	0.471
Zhou's	0.644	0.544	0.600	0.408	0.464	0.408	**0.755**	0.652	0.470
Mean	0.358	0.272	0.318	0.282	0.298	0.279	0.497	0.385	0.261

Table 3: Pearson correlation coefficients between expert assessment for **Content preservation** and content-based automatic metrics (using source sentences as references)

For style strength, system-level correlations between expert judgments and two style classifiers are computed and shown in Table 4. TextCNN has a higher correlation with expert judgments than fastText in general, it is consistent with the results of segment-level correlation for style strength in Table 6.

For fluency, system-level correlations between experts judgments and content-based automatic metrics are given in Table 5. For 5 out of 9 systems, expert judgments correlate best with BERTScore. However, for segment-level correlations of fluency in Table 6, COMET has the highest value. Overall, for fluency, expert assessments have a low degree of correlation or even no correlation with content-based automatic metrics.

Next, results of using human references as references used in content-based automatic metrics are introduced. System-level correlations between expert judgments for content preservation and metrics are shown in Table 7, BERTScore still has the highest correlation. Segment-level correlations between experts' judgments of

	TextCNN	fastText
BART large + SC + BLEU	**0.401**	0.287
Trained with High-Quality Generated Pairs	**0.408**	0.326
IBT + SC + BLEU + BLEURT	0.262	**0.293**
DURL	**0.612**	0.556
Bi-Directional Formality Transfer	0.316	**0.345**
NMT-Combined	**0.252**	0.198
Human references	**0.55**	0.243
StyIns	**0.713**	0.595
Zhou's	**0.582**	0.403
Mean	0.455	0.367

Table 4: Pearson correlation coefficients between expert assessment for **style** and style classifiers

	BLEU	METEOR	WMD	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	BLEURT	COMET
BART large + SC + BLEU	0.001	0.083	0.084	0.011	0.018	0.007	0.025	**0.114**	0.076
Trained with High-Quality Generated Pairs	0.029	0.025	0.008	0.055	0.061	0.055	**0.083**	0.080	0.044
IBT + SC + BLEU + BLEURT	0.082	0.032	0.084	0.084	0.084	0.081	**0.403**	0.088	0.109
DURL	0.425	0.469	0.535	0.344	0.482	0.349	0.56	**0.584**	0.46
Bi-Directional Formality Transfer	0.068	0.054	**0.117**	0.074	0.07	0.074	0.007	0.004	0.003
NMT-Combined	0.142	0.136	0.003	0.129	**0.161**	0.144	0.139	0.125	0.081
Human references	0.048	0.023	0.042	0.028	0.063	0.025	**0.166**	0.017	0.006
StyIns	0.322	0.19	0.253	0.18	0.112	0.18	**0.426**	0.41	0.359
Zhou's	0.398	0.437	0.389	0.164	0.278	0.164	**0.529**	0.527	0.234
Mean	0.168	0.161	0.168	0.119	0.148	0.12	0.26	0.217	0.152

Table 5: Pearson correlation coefficients between expert assessment for **fluency** and content-based automatic metrics (using source sentences as references)

	BLEU	METEOR	WMD	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	BLEURT	COMET	TextCNN	fastText
Content preservation	0.344	0.275	0.268	0.225	0.229	0.223	**0.443**	0.391	0.160	-	-
Style strength	-	-	-	-	-	-	-	-	-	**0.319**	0.272
Fluency	0.066	0.042	0.062	0.004	0	0	0.182	0.138	**0.212**	-	-

Table 6: Kendall's tau between expert assessments for all output sentences generated by 9 systems (8 text transfer systems + human references) and automatic metrics (for content-based metrics, source sentences are used as references)

output sentences from all systems and metrics are listed in Table 9. COMET has the best segment-level correlation for content preservation, which is different from the result of using source sentences as references in metrics. For fluency, BERTScore has the highest system-level correlation and COMET has the best segment-level correlation, which is consistent with the results of using source sentences as references in metrics. Nevertheless, when using human references as references in metrics, both segment-level correlations and system-level correlations are lower compared to using source sentences, for both content preservation and fluency.

As it is shown that there are differences in correlations for content preservation and fluency between using source sentences and human references as references used in content-based automatic metrics, it is natural to take a closer look at the differences. When using human references as references in metrics, most correlations are lower than the ones calculated when source sentences are used as references. Since content-based automatic metrics have a low degree of correlations with expert judgments for fluency, only differences in content preservation are studied. Absolute differences between Table 3 (except for *Human references* and Mean) and Table 7 (except for Mean) are shown in Figure 2. For all content-based metrics, the mean correlation difference varies from 0 to 0.3. Among all systems, BERTScore has the greatest difference in general and COMET has no difference, as COMET requires both source sentence and reference to calculate. For segment-level correlations (kendall's tau), the general difference in content preservation is consistent that using human references as references are lower than using source sentences, which can be seen in Figure 3. It could be the result of only showing the source sentences in human evaluation.

	BLEU	METEOR	WMD	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	BLEURT	COMET
BART large + SC + BLEU	0.113	0.101	0.176	0.156	0.133	0.164	**0.251**	0.149	0.182
Trained with High-Quality Generated Pairs	0.103	0.08	**0.162**	0.091	0.041	0.089	0.096	0.059	0.114
IBT + SC + BLEU + BLEURT	0.021	0.059	0.12	0.066	0.034	0.06	**0.138**	0.053	0.116
DURL	0.5	**0.709**	0.413	0.368	0.297	0.351	0.629	0.643	0.679
Bi-Directional Formality Transfer	0.049	0.063	0.034	0.075	0.029	0.082	**0.131**	**0.131**	0.063
NMT-Combined	0.037	0.044	0.086	0.013	0.027	0.008	**0.216**	0.198	0.158
StyIns	0.297	0.215	0.346	0.241	0.266	0.257	0.38	0.441	**0.471**
Zhou's	0.283	0.337	0.281	0.210	0.220	0.215	0.434	**0.520**	0.470
Mean	0.175	0.201	0.202	0.152	0.131	0.153	**0.284**	0.274	0.282

Table 7: Pearson correlation coefficients between expert assessment for **content preservation** and content-based automatic metrics (using human references as references)

	BLEU	METEOR	WMD	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	BLEURT	COMET
BART large + SC + BLEU	0.015	0.143	0.059	0.067	0.133	0.123	0.037	0.047	0.076
Trained with High-Quality Generated Pairs	0.096	0.061	0.116	0.07	0.033	0.032	0.023	0.104	0.044
IBT + SC + BLEU + BLEURT	0.07	0.107	0.133	0.091	0.088	0.113	0.187	0.042	0.109
DURL	0.283	0.469	0.238	0.193	0.176	0.177	0.398	0.366	0.46
Bi-Directional Formality Transfer	0.038	0.068	0.064	0.011	0.003	0.001	0.079	0.139	0.003
NMT-Combined	0.078	0.015	0.132	0.101	0.026	0.08	0.076	0.006	0.081
StyIns	0.19	0.2	0.248	0.222	0.258	0.225	0.282	0.192	0.359
Zhou's	0.135	0.133	0.113	0.019	0.055	0.03	0.152	0.227	0.234
Mean	0.113	0.149	0.138	0.097	0.097	0.098	0.154	0.140	0.171

Table 8: Pearson correlation coefficients between expert assessment for **fluency** and content-based automatic metrics (using human references as references)

	BLEU	METEOR	WMD	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	BLEURT	COMET
Content preservation	0.144	0.15	0.129	0.135	0.125	0.142	0.227	0.221	**0.237**
Fluency	0.111	0.125	0.109	0.121	0.111	0.124	0.189	0.163	**0.215**

Table 9: Kendall's tau between expert assessments for all output sentences generated by 8 systems and content-based metrics (human references are used as references)

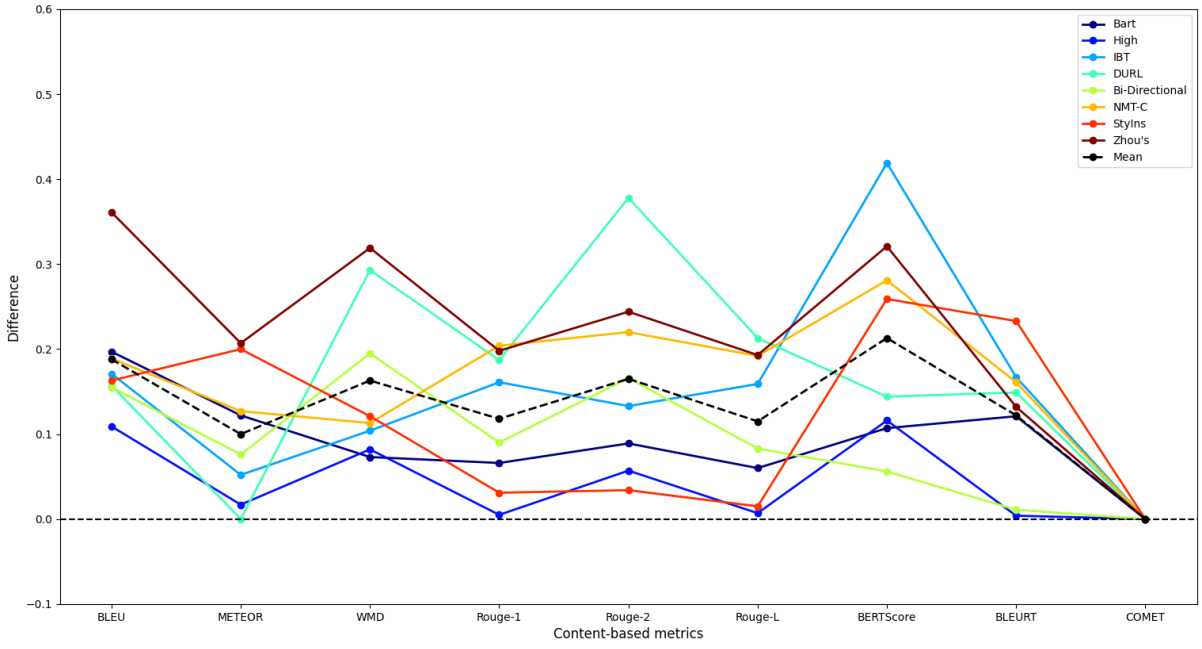

Figure 2: Differences in Pearson correlation coefficients for content preservation between using source sentences and human references as references used in metrics

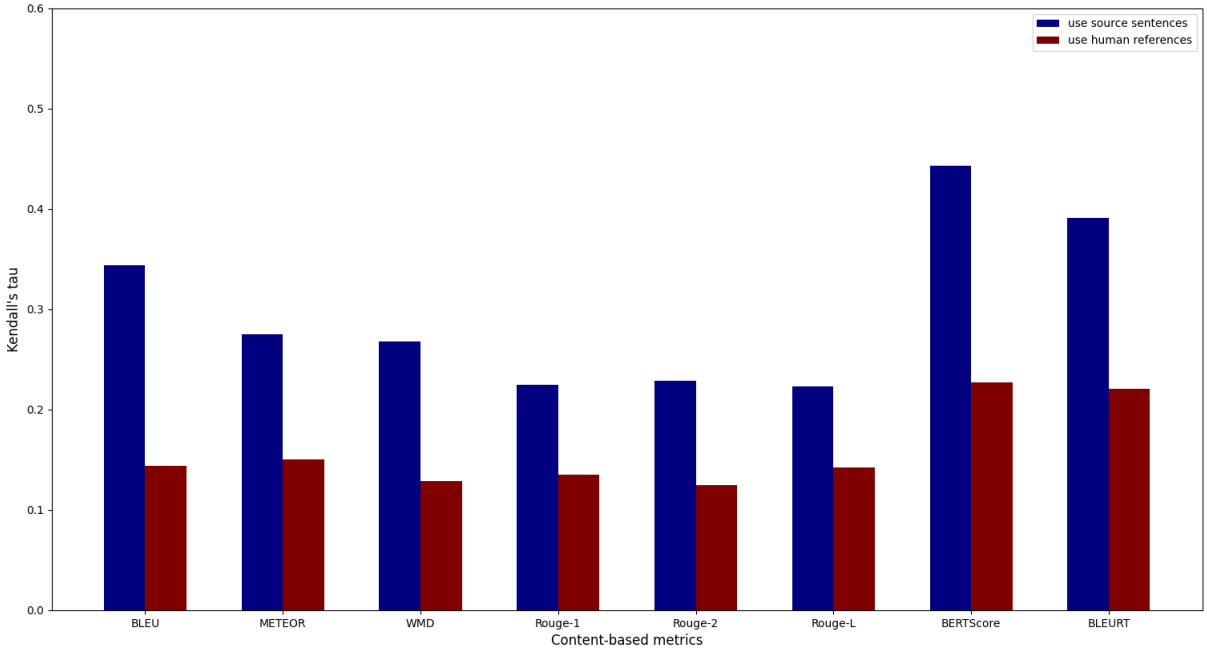

Figure 3: Differences in Kendall's tau for content preservation between using source sentences and human references as references used in metrics

4.2 RQ2

RQ2: What is a feasible, implementable setting for human evaluation of generated text, also with respect to observations in the current literature?

A brief introduction containing purposes, size, approximate completion time, requirements of the evaluation and contact information are presented in the beginning. Next, task guidelines including task definitions, criteria definitions For each output sentence being evaluated, the corresponding source sentence is shown as well. Three basic criteria should be assessed at least: (1) style strength: the amount that the output sentence fits the target style; (2) content preservation: the amount of content in the output sentence that remains unchanged compared to the corresponding source sentence; (3) fluency: the possibility of the output sentence written by a native speaker. From a practical point of view, it could be suitable to have an overall evaluation as well, as it can be used to see the overall performance intuitively. Clear definitions or concrete implementation of criteria are provided to annotators. Examples of evaluation of each transfer direction are shown, supplementary explanations are provided as well.

For measurements, as the continuous scale is more reliable than discrete Likert scales [39], continuous scale is implemented by, for example, using slider questions. Each criterion is equipped with a slider with labels indicating the range of scale.

For annotators, according to Freitag et al.[11], it is not reliable to only have crowd workers evaluation, as it has a low correlation with Multidimensional Quality Metrics based evaluation and even some automatic metrics (embedding-based) exceed the performance of crowd workers. But as van der Lee et al. [47] suggest, the number of participants of an evaluation should be sufficiently large, and it is not practical to get a large number of experts to evaluate. Therefore, both small-scale expert evaluation and large-scale crowd workers evaluation should be set in an ideal situation. The number of annotators and their backgrounds should be included. Each generated sentence is supposed to be evaluated at least twice for qualitative analysis, relevant Inter-annotator agreement (IAA) is reported [47].

In our experiment, 80 source sentences are divided into 4 parts, each of them contains 20 source sentences, the corresponding evaluations are contained in a survey, hence there are 4 surveys in total. Each survey is sent to two experts, which results in two annotations for each system output sentence. For each survey, Pearson's correlation is calculated as IAA, kappa are not used as IAA since the data is continuous and not categorical. Results in Table 10 indicate the experts understand the tasks uniformly, the guidelines are clear and the evaluation task is reproducible to a certain degree.

	Survey 1	Survey 2	Survey 4	Survey 5
Pearson's cor	0.70	0.68	0.72	0.66

Table 10: Inter-Annotator Agreements for all surveys

4.3 RQ3

RQ3: According to human assessment, which of the existing text style transfer system works best?

As mentioned before, output sentences generated by eight text style transfer systems are evaluated by experts, as well as the corresponding human references (which is seen as another system here). The mean values of scores given by experts of all sentences generated by a system for each criterion are shown in Table 11. It can be seen in Figure 4 that for content preservation, *Trained with High-Quality Generated Paires* [23] has the best result. For both style and fluency, *BART large + SC + BLEU* [24] has the best performance. It is interesting to see several text style transfer systems outperform humans (compared to assessment of human references).

	Content preservation	Style	Fluency
BART large + SC + BLEU	86.5	82.7	87.8
Trained with High-Quality Generated Pairs	92.4	76.3	83.3
IBT + SC + BLEU + BLEURT	85.2	80.1	86.0
DURL	47.6	46.7	37.9
Bi-Directional Formality Transfer	90.7	76.9	84.9
NMT-Combined	84.7	70.2	77.3
Human references	73.6	82.3	82.4
StyIns	50.5	51.1	38.6
Zhou's	50.9	47.2	45.1

Table 11: Experts assessment scores

5 Conclusions

In this thesis, we explore the common evaluation methods of natural language generation tasks, in particular for text style transfer for both automatic evaluation and human evaluation. Several text style transfer systems with a variety of architectures and two pre-trained models commonly used in text transfer systems are studied. Formality transfer, as one kind of text style transfer, is the main focus of the experiment. 8 style transfer systems with either supervised methods or unsupervised methods are selected for generating sentences of the formality transfer task. There are 3 criteria for the assessment, namely content preservation, style and fluency. For automatic evaluation of the text transfer systems, 7 content-based metrics (ROUGE has 3 variations, so 9 metrics in total) are used to evaluate content preservation and 2 style classifier are applied to assess how the sentences fit the target style. Expert evaluation is conducted as well, to evaluate the performances of not only text style transfer systems but also automatic metrics.

There are key findings of this project. First, based on expert judgments, among all text transfer systems evaluated, *Trained with High Quality Gnerated Paires* [23] works best in content preservation, and the system

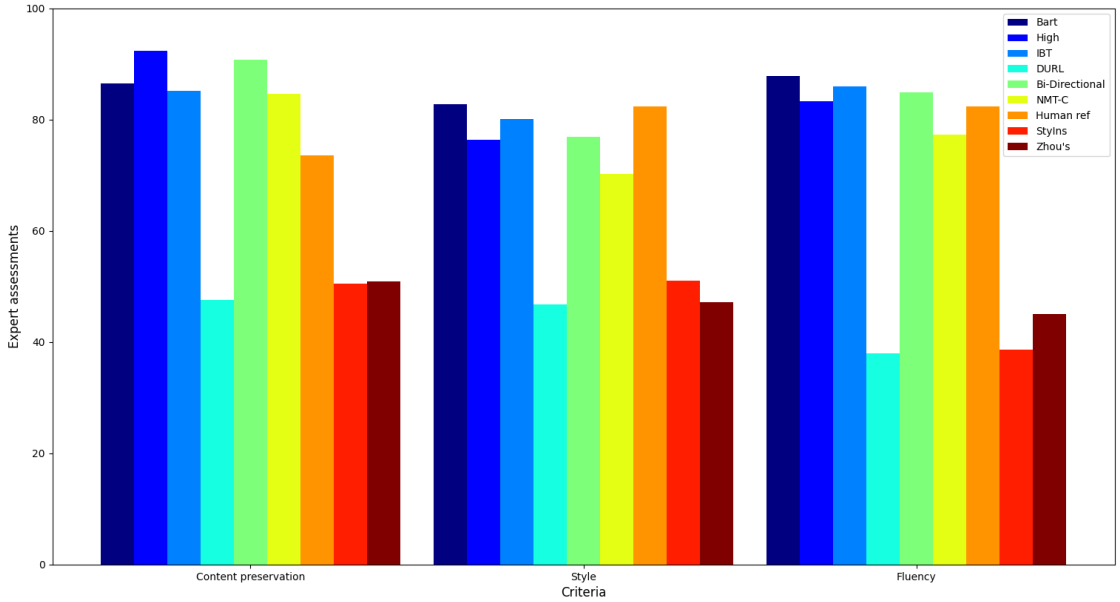

Figure 4: Expert assessment for 8 text style transfer systems and human references

BART large + SC + BLEU [24] has the best performance in style and fluency. Second, except for COMET, there are differences in correlations between using source sentences as references used in content-based metrics and using human references as references, for content preservation, the mean differences of 8 text style transfer system for 8 metrics varies from 0.1 to 0.3. Third, when using source sentences as references in content-based automatic metrics, for most of the systems, both content preservation and fluency correlate best with BERTScore, but in segment-level, fluency correlates best with COMET. Fourth, when using human references as references, content preservation correlates best with BERTScore in system-level and COMET in segment-level, fluency correlates best with COMET.

A web-based interface for human evaluation of formality transfer is established, it can be reused by assessing different automatic outputs and it can be transformed to evaluate other text style transfer tasks. It provides the contribution to the setup of human evaluation of Natural Language Generation. In addition, the number of text style transfer systems evaluated in the expert evaluation is large, which is not done in the previous studies. One limitation of this project is that only expert evaluation is done for human evaluation due to time constraints. Ideally, crowd worker evaluation on the same data given to the experts should be included as well. Therefore, crowd worker evaluation on formality transfer with the same text transfer systems and automatic metrics is left for future research.

References

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[2] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[3] Eleftheria Briakou, Sweta Agrawal, Ke Zhang, Joel Tetreault, and Marine Carpuat. A review of human evaluation for style transfer. In *Proceedings of the 1st Workshop on Natural Language Generation,*

Evaluation, and Metrics (GEM 2021), pages 58–67, Online, August 2021. Association for Computational Linguistics.

[4] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch, 2011.

[5] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020.

[6] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy, July 2019. Association for Computational Linguistics.

[7] Jan Milan Deriu and Mark Cieliebak. Syntactic manipulation for generating more diverse and interesting texts. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 22–34, Tilburg University, The Netherlands, November 2018. Association for Computational Linguistics.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[9] Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. Fighting offensive language on social media with unsupervised text style transfer, 2018.

[10] James Forrest, Somayajulu Sripada, Wei Pang, and George Coghill. Towards making NLG a voice for interpretable machine learning. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 177–182, Tilburg University, The Netherlands, November 2018. Association for Computational Linguistics.

[11] Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. Experts, errors, and context: A large-scale study of human evaluation for machine translation, 2021.

[12] Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online, August 2021. Association for Computational Linguistics.

[13] Vrindavan Harrison and Marilyn Walker. Neural generation of diverse questions using answer focus, contextual and linguistic features, 2018.

[14] David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland, December 2020. Association for Computational Linguistics.

[15] J. Edward Hu, Abhinav Singh, Nils Holzenberger, Matt Post, and Benjamin Van Durme. Large-scale, diverse, paraphrastic bitexts via sampling and clustering. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 44–54, Hong Kong, China, November 2019. Association for Computational Linguistics.

[16] Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. Deep learning for text style transfer: A survey, 2021.

[17] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification, 2016.

[18] Chris Kedzie and Kathleen McKeown. A good sample is hard to find: Noise injection sampling and self-training for neural language generation models. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 584–593, Tokyo, Japan, October–November 2019. Association for Computational Linguistics.

[19] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[20] Svetlana Kiritchenko and Saif Mohammad. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[21] Philipp Koehn and Christof Monz. Manual and automatic evaluation of machine translation between European languages. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 102–121, New York City, June 2006. Association for Computational Linguistics.

[22] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 957–966. JMLR.org, 2015.

[23] Huiyuan Lai, Antonio Toral, and Malvina Nissim. Generic resources are what you need: Style transfer tasks without task-specific parallel training data, 2021.

[24] Huiyuan Lai, Antonio Toral, and Malvina Nissim. Thank you bart! rewarding pre-trained models improves formality style transfer. *CoRR*, abs/2105.06947, 2021.

[25] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[26] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[27] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[28] Xiao Li, Guanyi Chen, Chenghua Lin, and Ruizhe Li. DGST: a dual-generator network for text style transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7131–7136, Online, November 2020. Association for Computational Linguistics.

[29] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[30] Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. A dual reinforcement learning framework for unsupervised text style transfer, 2019.

[31] Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy, August 2019. Association for Computational Linguistics.

[32] Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881, Online, July 2020. Association for Computational Linguistics.

[33] Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. The first multilingual surface realisation shared task (SR'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[34] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[35] Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 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)*, pages 495–504, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[36] Yusuke Mori, Hiroaki Yamane, Yusuke Mukuta, and Tatsuya Harada. Toward a better story end: Collecting human evaluation with reasons. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 383–390, Tokyo, Japan, October–November 2019. Association for Computational Linguistics.

[37] Xing Niu, Sudha Rao, and Marine Carpuat. Multi-task neural models for translating between styles within and across languages, 2018.

[38] Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[39] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. RankME: Reliable human ratings for natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 72–78, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[40] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

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

[42] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[43] Sudha Rao and Joel Tetreault. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer, 2018.

[44] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020. Association for Computational Linguistics.

[45] Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. Bleurt: Learning robust metrics for text generation, 2020.

[46] Anastasia Shimorina. Human vs automatic metrics: on the importance of correlation design, 2021.

[47] Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan, October–November 2019. Association for Computational Linguistics.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[49] Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators, 2019.

[50] Xiaoyuan Yi, Zhenghao Liu, Wenhao Li, and Maosong Sun. Text style transfer via learning style instance supported latent space. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3801–3807. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[51] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.

[52] Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun. Adversarially regularized autoencoders, 2018.

[53] Chulun Zhou, Liangyu Chen, Jiachen Liu, Xinyan Xiao, Jinsong Su, Sheng Guo, and Hua Wu. Exploring contextual word-level style relevance for unsupervised style transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7135–7144, Online, July 2020. Association for Computational Linguistics.