Neural Text Rewriting

Lai, Huiyuan

DOI:
10.33612/diss.910519765

CHAPTER 5

Generic Resources for Unsupervised Style Transfer

In this chapter we propose a novel approach to text style transfer that leverages generic resources, and without using any task-specific parallel (source–target) data outperforms existing unsupervised approaches on the two most popular tasks: formality transfer and polarity swap. In practice, we adopt a multi-step procedure which builds on a generic pretrained sequence-to-sequence model (BART). First, we strengthen the model's ability to *rewrite* by further pretraining BART on both an existing collection of generic paraphrases, as well as on synthetic pairs created using a general-purpose lexical resource. Second, through an iterative back-translation approach, we train two models, each in a transfer direction, so that they can provide each other with synthetically generated pairs, dynamically in the training process. Lastly, we let our best resulting model generate static synthetic pairs to be used in a supervised training regime. Besides novel methodology and state-of-the-art results, a core contribution of this chapter is a reflection on the nature of the two tasks we address, and how their differences are highlighted by their response to our approach.

Chapter adapted from:

Lai, H., Toral, A., and Nissim, M. (2021a). Generic resources are what you need: Style transfer tasks without task-specific parallel training data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4241–4254, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics

5.1 Introduction

In the recent tradition of text style transfer, formality and polarity swap have attracted growing interest in the Natural Language Generation (NLG) community (Zhang et al., 2018; Luo et al., 2019; Wu et al., 2019; Yi et al., 2020; Zhou et al., 2020). Generally, these two tasks are conflated in the literature under the general *style transfer* label and addressed with the same methods, but we find this an oversimplification. As the examples of the two tasks shown in Table 5.1, formality transfer implies rewriting a formal sentence into its informal counterpart (or viceversa) while preserving its meaning. Polarity swap, instead, aims to change a positive text into a negative one (or viceversa); and while the general theme must be preserved, the meaning is by definition not maintained. In line with previous work, we also address both tasks in a similar way, but this is actually to unveil how their different nature affects modelling and evaluation.

Due to the general scarcity of parallel data, previous works often adopt unsupervised approaches, dubbed *unpaired* methods (Dai et al., 2019) since they do not rely on labelled parallel pairs. However, it has also been shown that best results, unsurprisingly, can be achieved if parallel training data (such as the formality dataset (Rao and Tetreault, 2018)) is available (Rao and Tetreault, 2018; Sancheti et al., 2020). For this reason, substantial work has gone into the creation of artificial training pairs through various methods (see Section 5.2); approaches using synthetic pairs are thus still considered unsupervised in the style transfer literature, since they do not use manually labelled data.

In this chapter, we explore how parallel data can best be derived and integrated into a general style transfer framework. To do so, we create pairs in a variety of ways and use them at different stages of our framework. A core aspect of our approach is leveraging generic resources to derive training pairs, both natural and synthetic. On the natural front, we use abundant data from a generic rewriting task: paraphrasing. As for synthetic data, we leverage a general-purpose computational lexicon using its antonymy relation to generate polarity pairs.

In practice, we propose a framework that adopts a multi-step procedure

Task	Style	Sentences
Formality Transfer	Informal	no different between ages if the mind is near to eachother
	Formal	There is no difference between ages if the intellect is similar.
Polarity Swap	Negative	bad service in these areas and really ruined our visit.
	Positive	good service in these areas and really made our visit.
Paraphrase	Source	The bank is coming up on your left.
	Target	You have the bank on the left side.
	Source	I guess I've always been pretty good with words.
	Target	I think narrating has always been my strong suit.

Table 5.1: Example samples for each task: formality transfer from dataset GYAFC (Rao and Tetreault, 2018), polarity swap from YELP (Li et al., 2018), and paraphrase from PARABANK 2 (Hu et al., 2019).

which builds upon a general-purpose pretrained sequence-to-sequence (seq2seq) model. First, we strengthen the model's ability to rewrite by conducting a second phase of pretraining on pairs derived from an existing collection of generic paraphrases, as well as on synthetic pairs created using the general-purpose lexical resource. Second, through an iterative back-translation (Hoang et al., 2018) approach, we train two models, each in a transfer direction, so that they can provide each other with synthetically generated pairs on-the-fly. Lastly, we use our best resulting model to generate static synthetic pairs, which are then used offline as parallel training data.

Contributions Using a pretrained seq2seq model (1) we achieve state-of-the-art results for the two most popular text style transfer tasks in the unsupervised setting, i.e. without task-specific parallel data. We show that (2) generic resources can be leveraged to derive parallel data for additional model pretraining, which boosts the model's performance substantially and that (3) an iterative back-translation setting where models in the two transfer directions are trained simultaneously is successful, especially if enriched with the reward learning. Additionally, we offer (4) a theoretical contribution over the nature of the two tasks: while they are usually treated as the same task, our results suggest that they could possibly be treated separately.

5.2 Related Work

Text style transfer is most successful if task-specific parallel data is available, as in the case of formality transfer (Rao and Tetreault, 2018). Like in most NLP tasks, pretrained models (PLMs) have been shown to provide an excellent base for finetuning in a supervised setting, as we have also shown in Chapter 4. However, parallel data for finetuning such PLMs for style transfer is scarce. In recent years, a substantial amount of work has gone into methods for creating artificial sentence pairs so that models can be trained in a supervised regime.

One way to do this is to artificially generate parallel data via the back-translation (Sennrich et al., 2016), so that training pairs are created on-the-fly during the training process itself (Zhang et al., 2018; Lample et al., 2019; Prabhumoye et al., 2018; Luo et al., 2019). In these systems, one direction's outputs and its inputs can be used as pairs to train the model of the opposite transfer direction.

Another common strategy is to use style-word-editing to explicitly separate content and style (Li et al., 2018; Xu et al., 2018; Wu et al., 2019; Lee, 2020). These approaches first detect relevant words in the source and then do operations like deleting, inserting and combining to create the pair's target. Back-transferring is often used to reconstruct the source sentence for training, so that pairs are also made on-the-fly.

More recently, Li et al. (2020a) propose a two-stage strategy of search and learning for formality transfer where they perform a simulated annealing search (Liu et al., 2020b) to obtain output sentences as pseudo-references, and then finetune GPT-2 (Radford et al., 2019) with the resulting pairs.

The methods above create task-specific artificial pairs, some using pre-crafted manual rules or templates. Particularly, it is not evident which strategy works best for creating parallel data, whether offline or on-the-fly, and the simultaneous advantage of both strategies has not been fully explored. In this chapter, we aim to leverage generic resources to create task-specific training pairs. Since we have also shown in the previous chapter that a seq2seq pretrained model (BART) outperforms an autoregressive language model (GPT-2) when task-specific parallel training data is

Dataset	Style	Paired Train	Paired Valid	Test	Unpaired Train	Unpaired Valid
GYAFC [F&R]	Informal	51,967	2,788	1,332	N/A	N/A
	Formal	51.967	2,247	1,019	N/A	N/A
YELP	Negative	N/A	N/A	500	177,218	2,000
	Positive	N/A	N/A	500	266,041	2,000
PARABANK 2	Source	1,132,289	N/A	N/A	N/A	N/A
	Target	1,132,289	N/A	N/A	N/A	N/A

Table 5.2: Dataset Statistics.

available, we use BART as the generic base model here; we enrich it with iterative back-translation to create training pairs on-the-fly. We also explore the advantage of further pretraining by creating pairs through generic resources, as well as the benefits of a final training using generated pairs.

5.3 Tasks, Datasets, and Evaluation

This section begins by presenting the two tasks we consider and their corresponding datasets. In particular, we will compare how these two tasks differ and elicit the motivation for our approach. Subsequently, we briefly introduce the evaluation methods.

5.3.1 Tasks and Datasets

The task of style transfer is generally defined as the conversion of a text written in a given style to approximately the same text in a different style: style should be changed while preserving the original "content". We focus on the two most popular tasks, namely *formality transfer* and *polarity swap*, and use the two standard available datasets: GYAFC (F&R domain) and YELP, introduced in Chapter 3. Data statistics are in Table 5.2.

Although these two tasks have been conflated in previous work as "style transfer", they are not exactly the same, which we hypothesise affects both their modelling and evaluation. More specifically, in polarity swap the actual content is not exactly preserved (the message is actually the opposite), rather

it's the general "theme/topic" that needs to be preserved. In formality transfer, instead, the "translation" happens really more at style level, and content needs to stay the same. This is evident if we look at examples in Table 5.1 (top two blocks). In YELP, we can see that the theme-related words are expected to stay while changing the polarity words. Therefore, although the two sentences refer to the same event/concept, they convey opposite meanings. On the contrary, in formality transfer, an informal text should be changed into a formal one, but the overall meaning should be preserved. In this sense, formality transfer can be seen much more as rewriting than polarity swap and can be conceived akin to the more general task of paraphrasing.

Leveraging this observation, we explore if paraphrase pairs can be used to make the model learn the basic task of "rewriting" in a first stage. The advantage of using paraphrases is the large amount of parallel data available. Specifically, we use PARABANK 2, a large-scale, diverse, collection of paraphrases (Hu et al., 2019). Given the different nature of the two tasks, we expect this strategy to help more formality transfer than polarity swap, since the latter is much less of a rewriting task than the former. In spite of the differences highlighted above, we approach both tasks within the same framework for two reasons: (i) to compare to previous works, which have treated the tasks as manifestations of the same "style transfer" task; but also (ii) to observe if and how the tasks respond differently to modelling and evaluation metrics.

5.3.2 Task Evaluation

As introduced in Chapter 3, the performance of text style transfer is commonly assessed on style strength and content preservation.

For style strength, using a pretrained style classifier is the most popular automatic evaluation strategy. Here, our classifiers have an accuracy of 92.6% and 98.1% on the test sets of GYAFC (F&R) and YELP, respectively.

For content preservation, n-gram-based matching metrics such as BLEU (Papineni et al., 2002) are most commonly used, but usually fail to recognise information beyond the lexical level. To overcome such limitations, recent *learnable metrics* attempt to directly optimize correlation

with human judgments. Therefore, in addition to BLEU, which allows us to compare to previous work, we also use BLEURT and COMET[1]. Let us bear in mind that "content preservation" does not mean exactly the same thing for the two tasks that we consider (see Section 5.3.1), so that we might observe different reactions to different evaluation measures for the two tasks.

As overall score, for a direct comparison to previous work we compute the harmonic mean (HM) of style accuracy and BLEU.

5.4 Approach

We propose a framework that adopts a multi-step procedure on top of the large pretrained seq2seq model BART (Lewis et al., 2020).

Given a source sentence $\boldsymbol{x} = \{x_1, \cdots, x_n\}$ of length n with style s_1, the goal of text style transfer is to generate a sentence $\boldsymbol{y}$ with style s_2, preserving the source sentence's meaning in formality transfer or the source sentence's theme in polarity swap.[2] Formally, the objective is to minimize the following negative log likelihood:

$$L_\phi = -\Sigma_i \log(p(y_i|y_{1:i-1}, \boldsymbol{x}; \phi)) \tag{5.1}$$

where ϕ are the parameters of BART.

Our framework can be conceived as a pipeline, visualised in Figure 5.1. At the core of the framework are two BART models (model A and model B), one for each transfer direction. Since the main challenge in unpaired style transfer is that we cannot directly employ supervision (i.e. task-specific parallel training pairs), we explore and evaluate different ways of creating and using sentence pairs at different stages of the pipeline.

First, we strengthen the model's ability to rewrite by conducting a second phase of pretraining on natural pairs derived from an existing collection of generic paraphrases, as well as on synthetic pairs created using a general-purpose lexical resource (*Step 1*, Section 5.4.1).

[1] COMET is designed to also take input sentences into account, but our evaluations do not use the input setting because COMET training input and output are different languages

[2] In what follows, we use the term "content" in a more general way to refer to both cases.

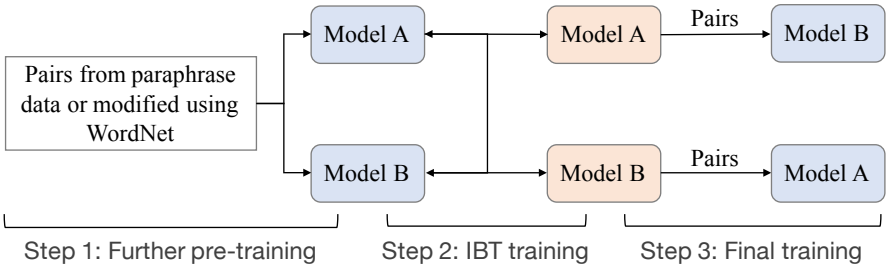

Figure 5.1: General overview of our pipeline.

Second, we use iterative back-translation with several reward strategies to train the two models in both transfer directions simultaneously; sentence pairs are created on the fly (*Step 2*, Section 5.4.2).

Third, we create high-quality synthetic pairs using our best systems from the previous step, to create a static resource of parallel data that can be used to train new transfer models (*Step 3*, Section in a supervised manner 5.4.3).

5.4.1 Further Pretraining: Learning to Rewrite

As hinted at in Section 5.3, style transfer can be seen as a specific way of *paraphrasing*. On the basis of this intuition, we hypothesise that generic paraphrase data, which already exists in much larger amounts than task-specific style transfer data, can be useful for text style transfer in terms of teaching the models the more generic task of "rewriting". For polarity swap, which is less of a rewriting task than formality transfer, as the meaning is reversed rather than preserved, we also create synthetic pairs using a general-purpose lexical resource.

Using the natural and the synthetic pairs we conduct a second phase of pretraining. We expect this strategy to help specifically with content preservation, which is known to be the most difficult part of style transfer, especially in an unsupervised setting (Sancheti et al., 2020).

Generic Training Pairs We use data from PARABANK 2 to make the model learn the basic task of "rewriting". We use this dataset in its entirety or filtered (models M1.1 and M1.2 in Table 5.3). In the first case, the whole of the

paraphrase pairs from PARABANK 2 are used to further pretrain the model. In the second case, we follow the rationale that not all pairs are equally relevant for our tasks, and selecting task-specific ones could be beneficial. For instance, while both PARABANK 2 pairs in Table 5.1 are good examples of rewriting, the second one is more meaningful in terms of formality transfer. Therefore, we train two binary style classifiers, one for formality and one for polarity, using TextCNN (Kim, 2014) on the training sets of GYAFC and YELP. These classifiers are then used to automatically select more strongly style-opposed pairs. The resulting filtered paraphrase subset $\boldsymbol{D}_p$ is such a set of pairs:

$$\boldsymbol{D}_p = \{(\boldsymbol{x}, \boldsymbol{y}) | (p(s_1|\boldsymbol{x}) + p(s_2|\boldsymbol{y}))/2 > \sigma\} \tag{5.2}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ constitute the sentence pair; $p(s_i|*)$ is the probability of a sentence being a style s_i, predicted by the style classifier, and σ is the threshold for data selection.[3]

Synthetic Pairs for Polarity Swap Due to the nature of polarity swap, we expect that even filtered paraphrases might not benefit polarity swap as much as formality. We therefore add another strategy to enhance polarity swap rewriting and create pairs for further pretraining exploiting a general-purpose lexical resource (model M1.3 in Table 5.3). Specifically, we use SentiWordNet (Baccianella et al., 2010) to obtain words' sentiment scores to detect the polarity of each word in the sentence. To maximise the quality of synthetic pairs, we select sentences that contain one polarity word only, and swap that one with its WordNet antonym (Miller, 1992). The new sentence is regarded as the target sentence corresponding to the original sentence.
The generic/filtered/synthetic pairs are used for a second phase of seq2seq pretraining for BART. Examples of these pairs are in Appendix B.1.

5.4.2 Iterative Back-translation: Pairs on-the-fly

After further pretraining BART, we use iterative back-translation to train two models, each in a transfer direction, so that they can provide one another with synthetically generated pairs on-the-fly. We obtain pseudo-parallel data

[3] σ = 0.85 in our experiments.

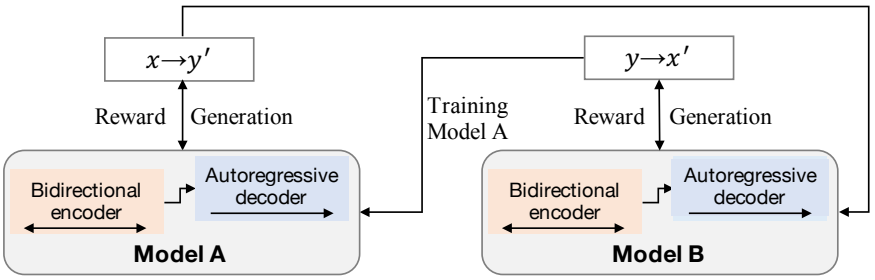

Figure 5.2: General overview of IBT training.

via back-transfer: the outputs of one direction are used to provide the supervision to train the model of the opposite direction (Figure 5.2). To explicitly guide the model to preserve the content and to apply the target style, we add content and style rewards in a reinforcement learning fashion (models M2.* in Table 5.3).

Rewarding Style Strength As introduced in Chapter 4.3.2, to provide an explicit signal to teach the model to change the sentence's style, a style classification (SC) reward is used to push the model to change the sentence into the target style. For this SC reward, which evaluates how well the transferred sentence $\boldsymbol{y}'$ matches the target style, we reuse the style classifier trained for selecting paraphrase data (Section 5.4.1). The SC's confidence in each transfer direction is

$$p(s_i|\boldsymbol{y}') = \text{softmax}_i(\text{TextCNN}(\boldsymbol{y}', \theta)) \tag{5.3}$$

where $s_{i\in\{1,2\}}$ represent source and target styles respectively. θ are the parameters of the classifier, fixed during the training of the transferred model. Formally, the reward is

$$R_{sc} = \lambda_{sc}[p(s_2|\boldsymbol{y}') - p(s_1|\boldsymbol{y}')] \tag{5.4}$$

where $\boldsymbol{y}'$ is the generated target sentence sampled from the distribution of model outputs at each decoding time step.

We apply the SC reward in two ways: in the supervised training process using pseudo-parallel data (SC0); and in the process of *generating* pseudo-parallel data itself (SC1). For the latter, we generate text in the target style

by sampling the distribution of model outputs, while at the same time using the SC reward to feed back its corresponding style signals to the model.

Rewarding Content Preservation Similar to Chapter 4.3.2, we use a BLEU-based reward to augment models in terms of content aspect, formulated as follows:

$$R_{bleu} = \lambda_{bleu}[\text{BLEU}(\boldsymbol{y}^s_{s_i}, \boldsymbol{x}) - \text{BLEU}(\boldsymbol{y}^g_{s_i}, \boldsymbol{x})] \tag{5.5}$$

where $\boldsymbol{y}^s_{s_i}$ is the generated sentence in target style s_i sampled from the distribution of model outputs at each time step in decoding, and $\boldsymbol{y}^g_{s_i}$ is obtained by greedily maximizing the distribution.

Since new-generation metrics show promising results in evaluation (Section 5.3), we use BLEURT also as an alternative metric to BLEU in the reward strategy, expecting it might be better at measuring semantics at the sentence level. Formally, we formulate the BLEURT-based reward as

$$R_{bleurt} = \lambda_{bleurt}[\text{BLEURT}(\boldsymbol{y}^s_{s_i}, \boldsymbol{x})] \tag{5.6}$$

where $\boldsymbol{y}^s_{s_i}$ is the generated sentence in target style s_i sampled from the distribution of model outputs.

Gradients and Objectives We use the policy gradient algorithm (Williams, 1992) to maximize the expected reward of the generated sentence $\boldsymbol{y}^s$, whose gradient with respect to the parameters ϕ of the neural network model is estimated by sampling:

$$\nabla_\phi J(\phi) = E[R \cdot \nabla_\phi \log(P(\boldsymbol{y}^s|\boldsymbol{x}; \phi))] \tag{5.7}$$

where $\nabla_\phi J(\cdot)$ is the gradient of objective function $J(\cdot)$ with respect to model parameters ϕ, $E(\cdot)$ is the expectation, R is the reward of the sequence $\boldsymbol{y}^s$ that is sampled from the distribution of model outputs at each decoding time step. The overall objectives are the combination of the base model's loss (Eq. 5.1) and the policy gradient of rewards (Eq. 5.7) which are used to train our framework end-to-end.

5.4.3 Final Training: High-quality Pairs

As a final step, we let our best models generate pairs from the previous step to create a static resource of parallel data. We feed the system source sentences

randomly picked from the training sets and generate the corresponding sentences in the target style. We then select high-quality pairs using BLEURT and our style classifier. The resulting dataset $\boldsymbol{D}_h$ is a set of pairs:

$$\begin{aligned}\boldsymbol{D}_h = \{(\boldsymbol{x}, \boldsymbol{y}^{'}) | &\text{BLEURT}(\boldsymbol{x}, \boldsymbol{y}^{'}) > \sigma_c \\ &\text{and } (p(s_1|\boldsymbol{x}) + p(s_2|\boldsymbol{y}^{'}))/2 > \sigma_s\}\end{aligned} \tag{5.8}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}^{'}$ are the source sentence and generated sentence, respectively. $p(s_i|*)$ is the probability of a sentence being of style s_i as predicted by the style classifier, and σ is the threshold for data selection regarding content and style.[4]

Finally, these pairs are used in the opposite direction $y^{'} \rightarrow x$ to finetune the original BART with all reward strategies, so as to train new transfer models in a supervised way (model M3.1 in Table 5.3).

5.5 Experiments

All experiments are implemented atop Huggingface Transformers (Wolf et al., 2020), taking the BART base model (139M parameters) for our experiments. We train our framework using the Adam optimiser (Kingma and Ba, 2015) with the initial learning rate 1e-5. The batch size is set to 32. The final values λ for style and content rewards are both set to 1 based on validation results. Both WordNet and SentiWordNet are used from NLTK [5].

5.5.1 Main Results

Table 5.3 reports results for each step.

Results of Step 1 show that using paraphrase data benefits more formality transfer than polarity swap, confirming the latter is much less of a rewriting task than the former. Filtering paraphrases to a subset closer to the task (M1.2) substantially helps formality and yields some improvement in polar-

[4] σ_c = 0.15 and σ_s = 0.9 in our experiments.

[5] https://www.nltk.org/

Dataset	Formality Transfer (GYAFC)					Polarity Swap (YELP)				
Model	BLEURT	COMET	BLEU	ACC	HM	BLEURT	COMET	BLEU	ACC	HM
M0: Vanilla BART	-11.6	24.2	41.4	33.3	36.9	-38.8	-14.6	30.9	2.2	4.1
STEP 1: Further pretraining with paraphrase data										
M1.1: Whole dataset	1.2	20.9	42.0	35.7	38.6	-41.2	-28.2	17.9	4.0	6.5
M1.2: Subset	1.1	22.5	44.1	69.3	53.9	-34.7	-17.8	24.7	16.6	19.9
M1.3: Synthetic data	-	-	-	-	-	-32.1	-7.4	32.6	18.9	23.9
STEP 2: IBT + Rewards										
M2.1: Based on M0	-1.0	29.2	50.7	83.6	63.1	-22.9	-1.7	29.8	82.6	43.8
M2.2: Based on M1.2	**4.1**	31.8	55.3	**93.2**	**69.4**	-17.6	2.6	29.5	85.3	43.8
M2.3: Based on M1.3	-	-	-	-	-	-24.6	-3.5	30.2	88.4	45.0
M2.4: M2.2 w/o BLEURT	3.3	31.3	55.2	92.9	69.3	-18.7	0.1	28.5	86.0	42.8
M2.5: M2.2 w/o BLEU	4.1	32.0	55.1	92.5	69.1	**-14.9**	3.1	29.5	78.4	42.9
M2.6: M2.2 w/o SC0	2.4	**32.1**	54.4	92.8	68.6	-19.5	-1.6	28.6	88.1	43.2
M2.7: M2.2 w/o SC1	3.9	31.8	55.5	87.3	67.9	-17.6	3.9	33.1	50.0	39.8
STEP 3: Offline training (Model used: original BART + Rewards)										
M3.1: High-quality pairs	3.0	**32.1**	**56.0**	90.4	69.2	-18.3	**4.6**	**31.6**	**88.7**	**46.6**
M3.2: data used in M1.2	1.1	22.9	45.5	78.3	57.6	-33.8	-22.1	21.5	45.7	29.2

Table 5.3: Results for the different steps of the pipeline. Notes: (i) SC0 is the SC reward used in the supervised training process using pseudo-parallel data, and SC1 is used in the process of generating pseudo-parallel data; (ii) high-quality pairs are generated by M2.2 for formality transfer and M2.3 for polarity transfer.

ity. WordNet-derived synthetic pairs (M1.3) are definitely a better strategy for polarity.[6]

The first block of Step 2 confirms that further pretraining significantly improves performance on formality transfer (compare M2.2 with M2.1). This results in the best model for formality transfer. For polarity, instead, we see improvement from further pretraining only when using WordNet-based synthetic pairs (compare M2.2 with M2.3). Overall, in Step 2 we see that combining SC rewards and content-related rewards results in the best balance regarding content preservation and style strength.

In Step 3, we see that the model trained with high-quality synthetic pairs (M3.1) achieves the best overall performance on polarity swap. For compar-

[6]The WordNet-based strategy could in principle be used on its own to solve the polarity swap task with no learning involved, but results prove it insufficient: BLEURT: -0.475; COMET: -0.221; BLEU: 0.296; ACC: 0.206; HM: 0.243.

Model	Sentence	BLEU	ACC
Informal→Formal			
Source	So if you're set on that, that's the way to go!!	-	
M0	so if you're set on that, that's the way to go!!	41.7	0.0
M1.1	so if you want to do this, this is the way to go!	30.1	0.3
M1.2	If you want to do this, this is the way to go.	41.6	85.5
M2.1	So if you're set on that, that is the way to go.	76.3	17.9
M2.2	So, if you are set on that, then that is the way to go.	**88.4**	**88.0**
M3.1	So if you are set on that, that is the way to go.	54.1	61.7
M3.2	If you're on board, that's the way to go.	35.2	55.2
Positive→Negative			
Source	the staff are all super friendly and on top of there jobs.	-	
M0	the staff are all super friendly and on top of there jobs.	16.3	0.0
M1.1	all the staff are very friendly and they're doing their jobs well.	10.7	0.3
M1.2	the staff are all super friendly and on top of each same jobs.	14.9	0.0
M1.3	the staff are all super unfriendly and on top of there jobs.	15.1	**100**
M2.1	the staff are all super rude and on top of there jobs.	15.1	**100**
M2.2	the staff are all super rude and on top of there jobs.	15.1	**100**
M2.3	the staff are all super rude and on top of there jobs.	15.1	**100**
M3.1	the staff are not super friendly or on top of there jobs.	**32.0**	**100**
M3.2	the staff are so friendly and they're doing their jobs.	14.8	0.1

Table 5.4: Example outputs for the different steps of the pipeline and their corresponding evaluation results. Note that ACC represents style confidence here.

ison, we use the subset of paraphrase data as training pairs in place of the generated pairs, and see that performance is lower (M3.2).

5.5.2 Case Study

Table 5.4 shows example outputs of each step and their evaluation results. It is interesting to see the impact of paraphrase-based pretraining: for formality, in M1.1 and M1.2, the phrase "*if you want to do this*" is used in place of "*if you're set on that*". This rewriting ability can also be observed on the polarity swap ("*on top of there jobs*" → "*they're doing their jobs well*"; note also that using paraphrases seems to prompt better writing: "*there*" → "*their*", M1.1/M3.2, though this is not consistent throughout the models).

For formality, the quality of the output gradually improves in Step 2, with

M2.2 achieving the best performance on BLEU and style confidence (M2.2). In M3.2, trained on paraphrase pairs, we find nice variability again ("*if you're on board*"). For polarity, M1.3 (using WordNet-based synthetic pairs), swaps a polarity word with its antonym ("*friendly*" → "*unfriendly*"). In Step 2, the models are indeed changing the polarity of the sentence; finally, the model trained with high-quality pairs (M3.1) nicely changes "*and*" into "*or*" to get the right semantics (though it loses the correct form "*their*") and is scored best. Further exploration of combining generic and task-specific rewriting appears very promising for these tasks.

As an additional curiosity-driven qualitative assessment of the behaviour of our models, we probed the polarity swap models with *neutral* sentences. As a first example, we use "*the earth revolves around the sun.*" as the source sentence, and observe that the models in both transfer directions generate the same sentences as the input. With as input the neutral sentence "*there is a grocery store near my house.*", the model which transforms negative sentences into positive ones generates "*there is a great grocery store near my house.*" while into the other direction it generates "*there is no grocery store near my house.*" It is worth mentioning that all the training data comes from business reviews on YELP, and the first example is clearly outside that domain. For the second example, closer to the domain of YELP, the transformation proposed by the model is rather reasonable in terms of obtaining a positive ("*great grocery store*") or negative ("*no grocery store*") output. It is left to future research to investigate what it should mean to transform a neutral sentence into a positive/negative one, and how such a test can help to better understand the models' behaviour and the task itself.

5.5.3 Comparison to Other Systems

To put our results in perspective, we compare our best systems (M2.2 for formality transfer and M3.1 for polarity swap in Table 5.3) against the most recent and best-performing unsupervised style transfer systems. For formality transfer: UnsuperMT (Zhang et al., 2018); DualRL (Luo et al., 2019); StyIns (Yi et al., 2020); Zhou's (Zhou et al., 2020); TGLS (Li et al., 2020a). For polarity swap: Style-Transformer (Dai et al., 2019); DualRL (Luo et al., 2019);

Model	BLEURT	COMET	BLEU	ACC	HM
Formality Transfer					
Input Copy	-11.4	27.2	47.4	12.0	19.2
UnsuperMT (Zhang et al., 2018)	-66.5	-44.6	32.7	67.0	43.9
DualRL (Luo et al., 2019)	-58.9	-45.1	40.4	65.4	49.9
StyIns (Yi et al., 2020)	-39.5	-11.2	45.8	76.1	57.3
Zhou's (Zhou et al., 2020)	-45.4	-20.3	44.7	79.9	57.3
TGLS (Li et al. (2020a); $0 \rightarrow 1$)	-	-	60.3	-	
Ours (M2.2; lowercase)	**0.9**	**32.8**	**56.3**	**86.6**	**68.2**
Ours (M2.2; lowercase; $0 \rightarrow 1$)	-	-	**74.1**	-	-
Polarity Swap					
Input Copy	-38.3	-13.9	31.2	1.9	3.6
Style-Transformer (Dai et al., 2019)	-46.9	-26.9	28.2	85.7	42.4
DualRL (Luo et al., 2019)	-38.5	-20.2	27.8	89.4	42.4
StyIns (Yi et al., 2020)	-57.6	-39.0	25.0	**92.4**	39.4
Zhou's (Zhou et al., 2020)	-27.0	-5.1	30.2	86.5	44.8
DGST (Li et al., 2020b)	-42.1	-24.0	26.8	78.1	39.9
Ours (M3.1)	**-18.3**	**4.6**	**31.6**	88.7	**46.6**

Table 5.5: Comparison with other systems. Notes: (i) we lowercase the GYAFC texts for a fairer comparison to previous works, as they do so; (ii) we report our results on informal-to-formal ($0 \rightarrow 1$) alone to compare with Li et al. (2020a), who only transfer in this direction, and the score is taken directly from the original paper.

StyIns (Yi et al., 2020); Zhou's (Zhou et al., 2020); DGST (Li et al., 2020b). In addition, we add a simple baseline for both tasks that just copies the input as output.

As visible in Table 5.5, our models achieve the best overall performance on both tasks. For formality transfer, this is true in all evaluation metrics. For polarity swap, StyIns has the highest style accuracy, while our model is better on all other metrics. [7]

[7]Full evaluation results using four human references for polarity swap are in Appendix B.2

Tasks	#	BLEURT	COMET	BLEU
		COMET	BLEU	BLEURT
Formality Transfer	17	0.98 (p<0.01)	0.76 (p<0.01)	0.74 (p<0.01)
Polarity Swap	17	0.81 (p<0.01)	0.48 (p=0.26)	0.29 (p=0.05)

Table 5.6: Pearson correlation between evaluation metrics for content preservation over 17 systems in Table5.3 (excluding M1.3 and M2.3) and 5.5 (excluding model TGLS in formality transfer).

5.6 Reflections on Tasks and Evaluation

The strategy of making the model learn the basic task of "rewriting" in a first stage clearly benefits more formality transfer than polarity swap. This is not surprising, since the latter is not simply "rewriting a sentence in a different stlye"; rather, the task involves changing the meaning of a sentence to obtain its opposite polarity, and thus, broadly put, its meaning. The fact that polarity swap cannot be regarded as a "style change" task is also evident from evaluation. Rather than only using BLEU, we suggested to also use BLEURT and COMET, and this provides us with additional evidence. Specifically, from Table 5.6 we observe that BLEU has a high correlation with BLEURT/COMET for formality transfer but not for polarity swap.

To glean further insights into this difference, we leverage human judgments released by Li et al. (2018) for YELP and see how they correlate with the used metrics. We calculate the system-level Pearson correlation between the automatic evaluations and human judgment. Results show that while COMET and BLEURT highly correlate with human judgments, BLEU does so to a lesser extent, suggesting this might be a less strong measure to assess the goodness of polarity swap.[8] Intuitively, if a system does change the polarity it may still have a high n-gram overlap (high BLEU) while new-generation metrics do not have this problem. For formality transfer, this limitation of BLEU is not much of an issue, since the meaning is not altered. Nevertheless,

[8]Pearson's $r = .922$ for BLEURT, $r = .941$ for COMET, and $r = .901$ for BLEU. All $p < .001, N = 7$.

we suggest that the evaluation of style transfer and related tasks should use learned metrics whenever possible.

5.7 Conclusion

In this chapter we proposed an unsupervised approach that adopts a multi-step procedure based on the general-purpose pretrained seq2seq model BART.

Achieving state-of-the-art results on the two most popular "style transfer" tasks, we have shown the benefit of further pretraining using data derived from generic resources as well as the advantage of back-translation, paired with rewards, especially towards content preservation. We have also seen how leveraging paraphrases can enhance both variability and naturalness in the generated text.

Through experimental settings as well as the introduction of BLEURT and COMET as metrics for these tasks, we have also revealed and highlighted how the two tasks we addressed differ. Indeed, we show that they benefit from partially different modelling, and react differently to evaluation metrics, both key aspects to improve future modelling of these tasks.

So far, we have focused on monolingual (English) text style transfer. In the next chapter, we will consider the task of multilingual style transfer, and address it by leveraging a lightweight language and task adaptation training without task-specific parallel data in the target languages.