Few-shot Controllable Style Transfer for Low-Resource Multilinugal Settings

Anonymous ACL submission

Abstract

Style transfer is the task of rewriting an input sentence into a target style while approximately preserving its content. While most prior literature assumes access to large style-labelled corpora, recent work (Riley et al., 2021) has attempted "few-shot" style transfer using only 3-10 sentences at inference for extracting the target style. In this work we study a relevant low-resource setting: style transfer for languages where no style-labelled corpora are available. We find that existing few-shot methods perform this task poorly, with a strong tendency to copy inputs *verbatim*.

We push the state-of-the-art for few-shot style transfer with a new method modeling the stylistic difference between paraphrases. When compared to prior work using automatic and human evaluations, our model achieves 2-3x better performance and output diversity in formality transfer and code-mixing addition across seven languages. Moreover, our method is better able to control the amount of style transfer using an input scalar knob. We report promising qualitative results for several attribute transfer directions, including sentiment transfer, text simplification, gender neutralization and text anonymization, all *without retraining the model*. Finally we found model evaluation to be difficult due to the lack of evaluation datasets and metrics for many languages. To facilitate further research in formality transfer for Indic languages, we crowdsource annotations for 4000 sentence pairs in four languages, and use this dataset[1] to design our automatic evaluation suite.

1 Introduction

Style transfer is a natural language generation task in which input sentences need to be re-written into a target style, while preserving semantics. It has many applications such as writing assistance (Heidorn, 2000), controlling generation for attributes like simplicity, formality or persuasion (Xu et al., 2015; Smith et al., 2020; Niu and Carpuat, 2020), data augmentation (Xie et al., 2019; Lee et al., 2021), and author obfuscation (Shetty et al., 2018).

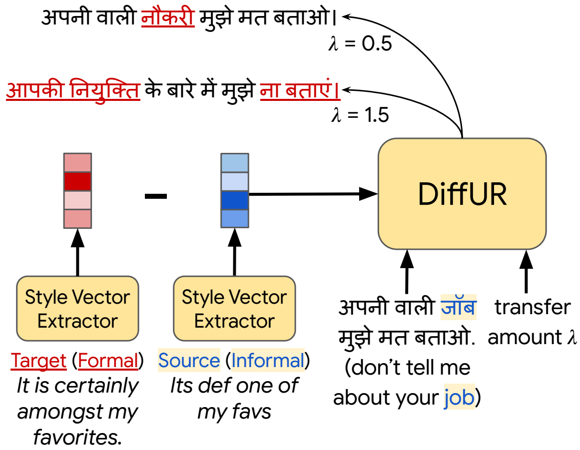

Figure 1: An illustration of our few-shot style transfer system during inference. Our model extracts style vectors from exemplar English sentences as input (in this case formal/informal sentences) and uses their vector difference to guide style transfer in other languages (Hindi). λ is used to control the magnitude of transfer: in this example our model produces more high Sanskrit words & honorifics (more formal) with higher λ.

Most prior work either assumes access to supervised data with parallel sentences between the two styles (Jhamtani et al., 2017), or access to large corpus of unpaired sentences with style labels (Prabhumoye et al., 2018; Subramanian et al., 2019). Models built are style-specific and cannot generalize to new styles during inference, which is needed for applications like real-time adaptation to a user's style in a dialog or writing application. Moreover, **access to a large unpaired corpus with style labels is a strong assumption**. Most standard "unpaired" style transfer datasets have been carefully curated (Shen et al., 2017) or were originally parallel (Xu et al., 2012; Rao and Tetreault, 2018). This is especially relevant in settings outside English, where NLP tools and labelled datasets are largely

[1]Dataset will be open-sourced on paper acceptance.

underdeveloped (Joshi et al., 2020). In this work, we take the **first steps** studying style transfer in **seven languages**[2] with nearly 1.5 billion speakers. Since no training data exists for these languages, we analyzed the current state-of-the-art in few-shot multilingual style transfer, the Universal Rewriter (UR) from Garcia et al. (2021). Unfortunately, we found it often copied input sentences verbatim (Section 3.1) *without transferring their style*.

We propose a simple inference-time trick of style-controlled translation through English, which improves the UR output diversity (Section 4.1). To further boost performance **we propose DIFFUR**,[3] an algorithm using the recent finding that paraphrasing leads to stylistic changes (Krishna et al., 2020). DIFFUR extracts edit vectors from paraphrase pairs, which are used to condition and train the model (Figure 2). On formality transfer and code-mixing addition, our best performing DIFFUR variant **significantly outperforms UR across all languages** (by 2-3x) using automatic & human evaluation. Besides better rewriting, our system is better able to **control the style transfer magnitude** (Figure 1). A scalar knob (λ) can be adjusted to make the output text reflect the target style (provided by exemplars) more or less. We also observe **promising qualitative results in several attribute transfer directions** (Section 6) including sentiment transfer, simplification, gender neutralization and text anonymization, all *without retraining the model* and using just 3-10 examples at inference.

Finally, we found it hard to precisely evaluate models due to the lack of evaluation datasets and style classifiers (often used as metrics) for many languages. To facilitate further research in Indic formality transfer, we crowdsource **formality annotations for 4000 sentence pairs in four Indic languages** (Section 5.1), and use this dataset to **design the automatic evaluation suite** (Section 5). In summary, our contributions provide an end-to-end recipe for developing and evaluating style transfer models and evaluation in a low-resource setting.

2 Related Work

Few-shot methods are a recent development in English style transfer, with prior work using variational autoencoders (Xu et al., 2020), or prompting large pretrained language models at inference (Reif et al., 2021). Most related is the state-of-the-art TextSETTR model from Riley et al. (2021), who use a neural style encoder to map exemplar sentences to a vector used to guide generation. To train this encoder, they use the idea that adjacent sentences in a document have a similar style. Recently, the **Universal Rewriter** (Garcia et al., 2021) extended TextSETTR to 101 languages, developing a joint model for translation, few-shot style transfer and stylized translation. This model is the only prior few-shot system we found outside English, and our main baseline. We discuss its shortcomings in Section 3.1, and propose fixes in Section 4.

Multilingual style transfer is mostly unexplored in prior work: a 35 paper survey by Briakou et al. (2021b) found only one work in Chinese, Russian, Latvian, Estonian, French. They further introduced XFORMAL, the first formality transfer *evaluation* dataset in French, Brazilian Portugese and Italian.[4] To the best of our knowledge, we are the first to study style *transfer* for the languages we consider. More related work from Hindi linguistics and on style transfer control is provided in Appendix B.

3 The Universal Rewriter (UR) model

We will start by discussing the Universal Rewriter (UR) model from Garcia et al. (2021), upon which our proposed DIFFUR model is built. The UR model extracts a style vector $\mathbf{s}$ from an exemplar sentence e, which reflects the desired target style. This style vector is used to style transfer an input sentence x. Consider $f_{\text{enc}}, f_{\text{dec}}$ to be encoder & decoder Transformers initialized with mT5 (Xue et al., 2021b), which are composed to form the model f_{ur}.

$$f_{\text{style}}(e) = \mathbf{s} = f_{\text{enc}}(\texttt{[CLS]} \oplus e)[0]$$
$$f_{\text{ur}}(x, \mathbf{s}) = f_{\text{dec}}(f_{\text{enc}}(x) + \mathbf{s})$$

where $\oplus$ is string concatenation, $+$ vector addition. f_{ur} is trained using the following objectives,

Learning Style Transfer by Exemplar-driven Denoising: To learn a style extractor, the Universal Rewriter uses the idea that two non-overlapping spans of text in the same document are likely to have the same style. Concretely, let x_1 and x_2 be two non-overlapping spans in mC4. Style extracted from one span is used to denoise the other,

$$\bar{x}_2 = f_{\text{ur}}(\text{noise}(x_2), f_{\text{style}}(x_1))$$
$$\mathcal{L}_{\text{denoise}} = \mathcal{L}_{\text{CE}}(\bar{x}_2, x_2)$$

[2]Indic (hi, bn, kn, gu, te), Spanish, Swahili.

[3]"**Diff**erence **U**niversal **R**ewriter", pronounced as *differ*.

[4]We do not use this data since it does not cover Indian languages, and due to Yahoo! L6 corpus restrictions for industry researchers (confirmed via authors correspondence).

where $\mathcal{L}_{\text{CE}}$ is the standard next-word prediction cross entropy loss function and noise$(\cdot)$ refers to 20-60% random token dropping and token replacement. This objective is used on the mC4 dataset (Xue et al., 2021b) with 101 languages. To build a general-purpose rewriter which can do translation as well as style transfer, the model is **additionally trained on two objectives**: (1) supervised machine translation using the OPUS-100 parallel dataset (Zhang et al., 2020), and (2) a self-supervised objective to learn effective style-controlled translation; more details in Appendix C.

During **inference** (Figure 1), consider an input sentence x and a transformation from style A to B (say *informal* to *formal*). Let S_A, S_B to be exemplar sentences in each of the styles (typically 3-10 sentences). The output y is computed as,

$$\mathbf{s}_A,\ \mathbf{s}_B = \frac{1}{N} \sum_{y \in S_A,\ S_B} f_{\text{style}}(y)$$

$$y = f_{\text{ur}}(x, \lambda(\mathbf{s}_B - \mathbf{s}_A))$$

where λ acts as a control knob to determine the magnitude of style transfer, and the vector subtraction helps remove confounding style information.[5]

3.1 Shortcomings of the Universal Rewriter

We experimented with the UR model on Hindi formality transfer, and noticed poor performance. We noticed that UR has a **strong tendency to copy sentences *verbatim*** — 45.5% outputs were copied exactly from the input (and hence not style transferred) for the best performing value of λ. The copying increase for smaller λ, making magnitude control harder. We identify the following issues:

1. **Random token noise leads to unnatural inputs & transformations:** The Universal Rewriter uses 20-60% uniformly random token dropping / replacement to noise inputs, which leads to ungrammatical inputs during training. We hypothesize models tend to learn grammatical error correction, which encourages verbatim copying during inference where fluent inputs are used and no error correction is needed. Moreover, token-level noise does not differentiate between content / function words, and cannot do syntactic changes like content reordering (Goyal and Durrett, 2020). Too much noise could distort semantics and encourage hallucination, whereas too little will encourage copying.

2. **Style vectors may not capture the precise style transformation:** The Universal Rewriter extracts the style vector from a single sentence during training, which is a mismatch from the inference where a *difference* between vectors is taken. Without taking vector differences at inference, we observe semantic preservation and overall performance of the UR model is much lower.[6]

3. **mC4 is noisy**: On reading training data samples, we noticed noisy samples with severe language identification errors in the Hindi subset of mC4. This has also been observed recently in Caswell et al. (2021), who audit 100 sentences in each language, and report 50% sentences in Marathi and 20% sentences in Hindi have the wrong language.

4. **No translation data for several languages:** We notice worse performance for languages which did not get parallel translation data (for the translation objective in Section 3). In Table 1 we see UR gets a score[7] of 30.4 for Hindi and Bengali, languages for which it got translation data. However, the scores are lower for Kannada, Telugu & Gujarati (25.5, 22.8, 23.7), for which no translation data was used. We hypothesize translation data encourages learning language-agnostic semantic representations needed for translation from the given language, which in-turn improves style transfer.

4 Our Models

4.1 Style-Controlled Backtranslation (+ BT)

While the Universal Rewriter model has a strong tendency to exactly copy input sentences while rewriting sentences in the same language (Section 3.1), we found it is an effective style-controlled *translation* system. This motivates a simple **inference-time** trick to improve model outputs and reduce copying — translate sentences to English (`en`) in a style-agnostic manner with a zero style vector $\mathbf{0}$, and translate back into the source language (`lx`) with stylistic control.

$$\mathbf{s}_A,\ \mathbf{s}_B = \frac{1}{N} \sum_{y \in S_A,\ S_B} f_{\text{style}}(y)$$

$$x^{\texttt{en}} = f_{\text{ur}}(\texttt{en} \oplus x, \mathbf{0})$$

$$\bar{x} = f_{\text{ur}}(\texttt{lx} \oplus x^{\texttt{en}}, \lambda(\mathbf{s}_B - \mathbf{s}_A))$$

[5] Garcia et al. (2021) also recommend adding the style vectors from the input sentence x, but we found this increased the amount of verbatim copying and led to poor performance.

[6] This difference possibly helps remove confounding information (like semantic properties, other styles) and focus on the specific style transformation. Since two spans in the same document will share aspects like article topic / subject along with style, we expect these semantic properties will confound the style vector space obtained after the UR training.

[7] Using the r-AGG style transfer metric from Section 5.5.

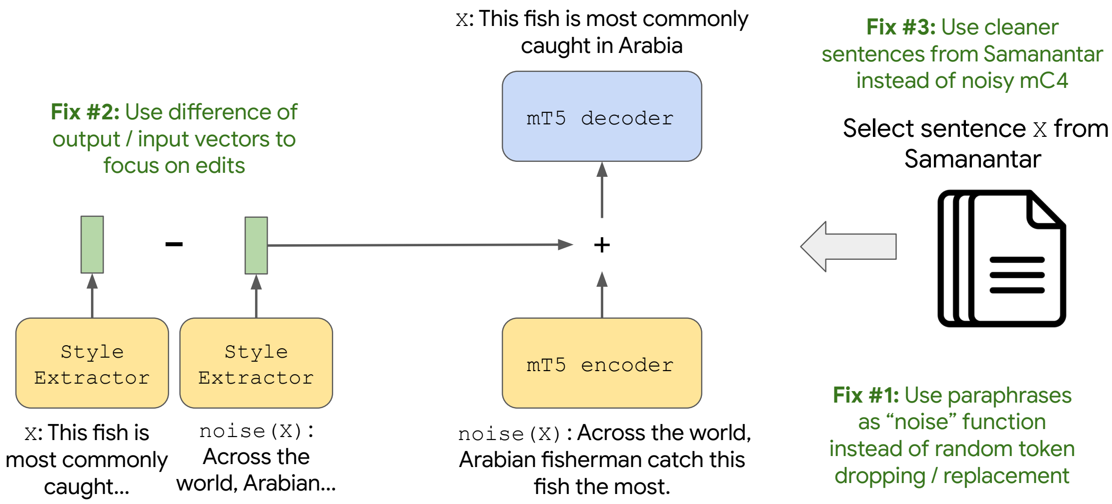

Figure 2: The DIFFUR approach (Section 4.2), with fixes to the shortcomings of the Universal Rewriter approach (Section 3.1) shown. Sentences are noised using paraphrasing, the style vector difference between the paraphrase & original sentence ("edit vector") is used to control denoising. See Figure 1 for the inference-time process.

where x is the input sentence, S_A, S_B are exemplars of the styles we want to transfer between, `en`, `lx` are language codes prepended to indicate the output language (Appendix C). Prior work has shown that backtranslation is effective for paraphrasing (Wieting and Gimpel, 2018; Iyyer et al., 2018) and style transfer (Prabhumoye et al., 2018).

4.2 Using Paraphrase Vector Differences for Style Transfer (DIFFUR)

While style-controlled backtranslation is an effective strategy, it needs two translation steps. This is 2x slower than UR, and semantic errors increase with successive translations. To learn effective style transfer systems needing only a single generation step we develop DIFFUR, a new few-shot style transfer training objective (overview in Figure 2). DIFFUR tackles the issues discussed in Section 3.1 using paraphrases and style vector differences.

Paraphrases as a "noise" function: Instead of using random token-level noise (issue #1 in Section 3.1), we paraphrase sentences to "noise" them during training. Paraphrasing modifies the lexical & syntactic properties of sentences, while preserving fluency and input semantics. Prior work (Krishna et al., 2020) has shown that paraphrasing leads to stylistic changes, and denoising can be considered a style re-insertion process.

To create paraphrases, we backtranslate sentences from the UR model[8] with no style control (zero vectors used as style vectors). To increase diversity, we use random sampling in both translation steps, pooling generations obtained using temperature values $[0.4, 0.6, 0.8, 1.0]$. Finally, we discard paraphrase pairs from the training data where the semantic similarity score[9] is outside the range $[0.7, 0.98]$. This removes backtransation errors (score < 0.7), and exact copies (score > 0.98).

Using style vector differences for control: To fix the training / inference mismatch for style extraction (issue #2 in Section 3.1), we propose using style vector differences between the output and input as the stylistic control. Concretely, let x be an input sentence and x_{para} its paraphrase.

$$\begin{aligned} \mathbf{s}_{\text{diff}} &= f_{\text{style}}(x) - f_{\text{style}}(x_{\text{para}}) \\ \bar{x} &= f_{\text{ur}}(x_{\text{para}}, \text{stop-grad}(\mathbf{s}_{\text{diff}})) \\ \mathcal{L} &= \mathcal{L}_{\text{CE}}(\bar{x}, x) \end{aligned}$$

where stop-grad$(\cdot)$ stops gradient flow through $\mathbf{s}_{\text{diff}}$, preventing the model from learning to copy x exactly. To ensure f_{style} extracts meaningful style representations, we fine-tune a trained UR model. Vector differences have many advantages,

1. Subtracting style vectors between a sentence and its paraphrase removes confounding features (like semantics) present in the vectors.
2. The vector difference focuses on the precise transformation that is needed to reconstruct the input from its paraphrase.
3. The length of $\mathbf{s}_{\text{diff}}$ acts as a proxy for the amount of style transfer, which is controlled using λ during inference (Section 3).

[8] Specifically, an Indic variant of the UR model is used, described in Section 4.3. Note it is not necessary to use UR for backtranslation, any good translation model can be used.

[9] Calculated using LaBSE, discussed in Section 5.3.

DIFFUR is related to neural editor models (Guu et al., 2018; He et al., 2020), where language models are decomposed into a probabilistic space of edit vectors over prototype sentences. We justify the DIFFUR design with ablations in Appendix G.1.

4.3 Indic Models (UR-INDIC, DIFFUR-INDIC)

To address the issue of no translation data (issue #4 in Section 3.1), we train Indic variants of our models. We replace the OPUS translation data used for training the Universal Rewriter (Section 3) with Samanantar (Ramesh et al., 2021), which is the largest publicly available parallel translation corpus for 11 Indic languages. We call these variants UR-INDIC and DIFFUR-INDIC. This process significantly up-samples the parallel data seen between English / Indic languages, and gives us better performance (Table 1) and lower copy rates, especially for languages with no OPUS translation data.

4.4 Multitask Learning (DIFFUR-MLT)

One issue with our DIFFUR-INDIC setup is usage of a stop-grad($\cdot$), to avoid verbatim copying from the input. This prevents gradient flow into the style extractor f_{style}, and as we see in Appendix H, a degradation of the style vector space. To prevent this from happening, we simply do multi-task learning between the original Universal Rewriter objective (Section 3) and our DIFFUR-INDIC objective, using an equal number of minibatches for each objective.

5 Evaluation

Automatic evaluation of style transfer is challenging (Pang, 2019; Mir et al., 2019; Tikhonov et al., 2019), and the lack of resources (such as evaluation datasets, style classifiers) make evaluation trickier for Indic languages. To tackle this issue, we first collect a small dataset of formality and semantic similarity annotations in four Indic languages (Section 5.1). We use this dataset to guide the design of an evaluation suite (Section 5.2-5.6). Since automatic metrics in generation are imperfect (Celikyilmaz et al., 2020), we complement our results with human evaluation (Section 5.7).

5.1 Indic Formality Transfer Dataset

Since no public datasets exist for formality transfer in Indic languages, it is hard to measure the extent to which automatic metrics (such as style classifiers) are effective. To tackle this issue, we build a dataset of **1000** sentence pairs in **each of four Indic languages** (Hindi, Bengali, Kannada, Telugu) with formality and semantic similarity annotations. We first style transfer held-out Samanantar sentences using our UR-INDIC + BT model (Section 4.1, 4.3) to create sentence pairs with different formality. We then asked three crowdworkers to 1) label the more formal sentence in each pair; 2) rate semantic similarity on a 3-point scale.

Our crowdsourcing is conducted on Task Mate,[10] where we hired native speakers from India with at least a high school education and 90% approval rating on the platform. To ensure crowdworkers understood "formality", we provided instructions following advice from professional Indian linguists, and asked two qualification questions in their native language. More details (agreement, compensation, instructions) are provided in Appendix E.4.

5.2 Transfer Accuracy (r-ACC, a-ACC)

Our first metric checks whether the output sentence reflects the target style. This is measured by an external classifier's predictions on system outputs. We use two variants of transfer accuracy: (1) Relative Accuracy (r-ACC): does the target style classifier score the output sentence *higher* than the input sentence? (2) Absolute Accuracy (a-ACC): does the classifier score the output *higher* than 0.5?

Building multilingual classifiers: Unfortunately, no large style classification datasets exist for most languages, preventing us from building classifiers from scratch. We resort to zero-shot cross lingual transfer techniques (Conneau and Lample, 2019), where large multilingual pretrained models are first fine-tuned on English classification data, and then applied to other languages at inference. We experiment with three such techniques, and find MAD-X classifiers with language adapters (Pfeiffer et al., 2020b) have the highest accuracy of 81% on our Hindi data from Section 5.1. However, MAD-X classifiers were only available for Hindi, so we use the next best XLM RoBERTa-base (Conneau et al., 2020) for other languages, which has 75%-82% accuracy on annotated data; details in Appendix E.1.

5.3 Semantic Similarity (SIM)

Our second evaluation criteria is semantic similarity between the input and output. Following recent recommendations (Marie et al., 2021; Krishna et al., 2020), we avoid n-gram overlap metrics like BLEU (Papineni et al., 2002). Instead, we use

[10] `https://taskmate.google.com`

LaBSE (Feng et al., 2020), a language-agnostic semantic similarity model based on multilingual BERT (Devlin et al., 2019). LaBSE supports 109 languages, and is the only similarity model we found supporting all the Indic languages in this work. We also observed LaBSE had greater correlation with our annotated data (Section 5.1) compared to alternatives; details in Appendix E.2.

Qualitatively, we found that sentence pairs with LaBSE scores lower than 0.6 were almost never paraphrases. To avoid rewarding partial credit for low LaBSE scores, we use a hard threshold[11] ($L = 0.75$) to determine whether pairs are paraphrases,

$$\textsc{sim}(x, y') = 1 \text{ if } \{\text{LaBSE}(x, y') > L\} \text{ else } 0$$

5.4 Other Metrics (LANG, COPY, 1-g)

Additionally, we measure whether the input and output sentences are in the same language (LANG), the fraction of outputs copied verbatim from the input (COPY), and the 1-gram overlap between input / output (1-g). High LANG and low COPY / 1-g (more diversity) is better; details in Appendix E.6.

5.5 Aggregated Score (r-AGG, a-AGG)

To get a sense of overall system performance, we combine individual metrics into one score. Similar to Krishna et al. (2020) we aggregate metrics as,

$$\textsc{agg}(x, y') = \textsc{acc}(x, y') \cdot \textsc{sim}(x, y') \cdot \textsc{lang}(y')$$

$$\textsc{agg}(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{x, y' \in \mathcal{D}} \textsc{agg}(x, y')$$

Where (x, y') are input-output pairs, and $\mathcal{D}$ is the test corpus. In other words, we measure the fraction of outputs which *simultaneously* transfer style, have a semantic similarity of at least L (our threshold in Section 5.3), and have the same language as the input. Depending on the variant of ACC (relative / absolute), we can derive r-AGG / a-AGG.

5.6 Evaluating Control (CALIB)

An ideal system should not only be able to style transfer sentences, but also control the *magnitude* of style transfer using the scalar input λ. To evaluate this, for every system we first determine a $\lambda_{\max}$ value and let $[0, \lambda_{\max}]$ be the range of control values. While in our setup λ is an unbounded scalar, we noticed high values of λ significantly perturb semantics (also noted in Garcia et al., 2021), with systems outputting style-specific n-grams unfaithful to the output. We choose $\lambda_{\max}$ to be the largest λ from the list $[0.5, 1.0, 1.5, 2.0, 2.5, 3.0]$ whose outputs have an average semantic similarity score (SIM, Section 5.3) of at least 0.75[12] with the validation set inputs. For each system we take three evenly spaced λ values in its control range, denoted as $\Lambda = [\frac{1}{3}\lambda_{\max}, \frac{2}{3}\lambda_{\max}, \lambda_{\max}]$. We then compute the **style calibration to** λ (CALIB), or how often does increasing λ lead to a style score increase? We measure this with a statistic similar to Kendall's τ (Kendall, 1938), counting concordant pairs in Λ,

$$\textsc{calib}(x) = \frac{1}{n} \sum_{\lambda_b > \lambda_a} \{\text{style}(y_{\lambda_b}) > \text{style}(y_{\lambda_a})\}$$

where x is input, $\textsc{calib}(x)$ is the average over all possible n $(= 3)$ pairs of λ values (λ_a, λ_b) in Λ.

5.7 Human Evaluation

Automatic metrics are usually insufficient for style transfer evaluation — according to Briakou et al. (2021a), 69 / 97 surveyed style transfer papers used human evaluation. We adopt the crowd-sourcing setup from Section 5.1, which was used to build our formality evaluation datasets. We presented 200 generations from each model and the corresponding inputs in a random order, and asked three crowdworkers two questions about each pair of sentences: (1) which sentence is more formal/code-mixed? (2) how similar are the two sentences in meaning? This lets us evaluate r-ACC, SIM, r-AGG, CALIB with respect to human annotations instead of classifier predictions; details in Appendix E.4.

6 Main Experiments

We evaluate models on (1) formality transfer; (2) increasing the amount of code-mixing with English. **Seven languages** with varying scripts and morphological richness are used for evaluation (`hi,es,sw,bn,kn,te,gu`). Note that no paired/unpaired data with style labels is used during training: models determine the target style at inference using **3-10 exemplars** sentences. For few-shot formality transfer, we use the English exemplars from Garcia et al. (2021). We follow their setup and use English exemplars to guide non-English transfer zero-shot. For code-mixing addition, we use Hindi/English code-mixed exemplars

[11] Roughly 73% pairs annotated as paraphrases (from dataset in Section 5.1) had $L > 0.75$. We experiment with different values of L in Appendix E.3 and notice similar trends.

[12] This threshold is identical to the value chosen for paraphrase similarity in Section 5.3. We experiment with more/less conservative thresholds in Appendix E.3.

Model	Hindi		Bengali		Kannada		Telugu		Gujarati	
	r-AGG	a-AGG	r-AGG	a-AGG	r-AGG	a-AGG	r-AGG	a-AGG	r-AGG	a-AGG
UR (2021)	30.4	10.4	30.4	7.2	25.5	8.0	22.8	8.4	23.7	5.0
UR-INDIC	58.3	18.6	65.5	22.3	61.3	17.8	59.8	19.9	54.0	10.7
UR + BT	54.2	17.8	55.6	16.9	39.8	11.9	38.4	11.6	46.3	10.4
UR-INDIC + BT	60.0	22.2	61.1	22.0	59.2	21.0	56.8	22.2	57.7	16.8
DIFFUR	71.1	22.9	72.7	25.2	69.2	29.1	69.4	27.1	0.4	0.2
DIFFUR-INDIC	72.6	24.0	75.4	24.3	73.1	29.3	71.0	27.1	36.0	13.0
DIFFUR-MLT	**78.1**	**32.2**	**80.0**	**35.0**	**80.4**	**39.4**	**79.8**	**37.9**	**75.0**	**33.1**

Table 1: Automatic evaluation of formality transfer in Indic languages. Note each proposed method (*-INDIC, +BT, DIFFUR) improves performance (AGG defined in Section 5.5), with a combination (DIFFUR-MLT) doing best.

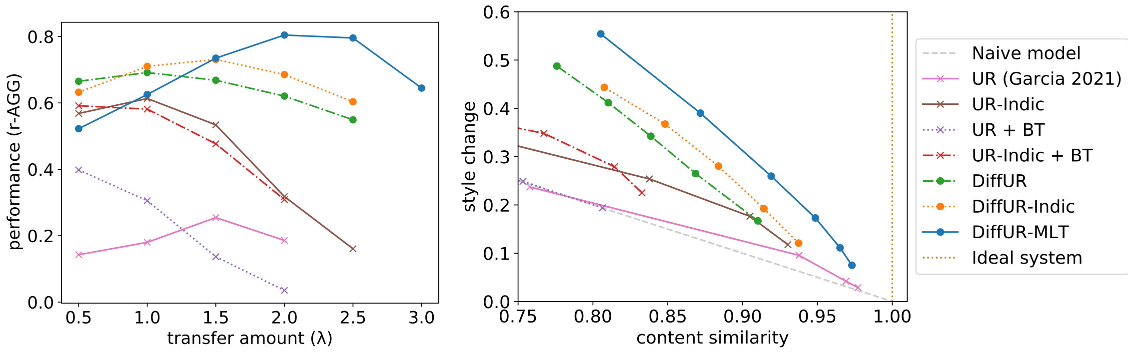

Figure 3: Variation in Kannada formality transfer with λ. In the *left* plot, we see DIFFUR-* models have consistently good overall performance with change in λ. In the *right* plot, we see the tradeoff between average style change and content similarity as λ is varied. Plots (such as DIFFUR-*) which stretch the Y-axis range, closer to the ideal system ($x = 1$) and away from the naive system ($x + y = 1$, akin to naive model in Krishna et al., 2020) are better.

Model	Swahili	Spanish
	r-AGG / a-AGG	r-AGG / a-AGG
UR (2021)	19.9 / 4.8	13.4 / 1.3
UR, BT	13.7 / 3.4	33.3 / 5.8
DIFFUR-MLT	**32.2 / 7.2**	**46.5 / 16.5**

Table 2: Automatic evaluation of formality transfer in Swahili and Spanish. DIFFUR-MLT performs best.

Model	ACC	SIM	AGG	CALIB	C-IN
UR (2021)	29.5	**87.2**	23.2	-	-
UR-INDIC	46.5	85.3	40.8	35.7	43.0
UR + BT	57.5	71.2	42.9	-	-
UR-INDIC + BT	65.0	77.8	52.4	24.0	40.3
DIFFUR	64.5	80.8	52.0	-	-
DIFFUR-INDIC	62.0	83.1	50.4	48.0	**54.5**
DIFFUR-MLT	**70.0**	80.8	**55.6**	**53.0**	**54.5**

Table 3: Human evaluation on Hindi formality transfer, measuring style accuracy (ACC), input similarity (SIM), overall score (AGG) and control with λ (CALIB, C-IN). Like Table 1, DIFFUR-MLT performs best.

Model	Hindi	Bengali
	ACC / SIM / AGG	ACC / SIM / AGG
UR (2021)	4.5 / **93.8** / 3.6	0.0 / **96.4** / 0.0
UR-INDIC,BT	18.5 / 79.2 / 15.3	18.0 / 68.3 / 12.7
DIFFUR-MLT,BT	**62.5** / 69.9 / **41.5**	**79.0** / 57.1 / **43.5**

Table 4: Human evaluation on code-mixing addition. DIFFUR-MLT+BT performs best (AGG), giving high style accuracy (ACC) without loss in similarity (SIM).

Model	CALIB	Model	CALIB
UR (2021)	29.2	DIFFUR	64.9
UR-INDIC	60.7	DIFFUR-INDIC	**69.6**
UR + BT	43.4	DIFFUR-MLT	69.0
UR-INDIC + BT	38.7		

Table 5: Evaluation of Hindi formality transfer magnitude control using λ. We find that DIFFUR-* are best at calibrating style change (CALIB) to input λ (metrics details in Section 5.6, more results in Appendix F).

in Devanagari (shown in Appendix D); more details of our training & evaluation setup in Appendix A.

Each proposed method improves over prior work, DIFFUR-MLT works best. We present our automatic evaluation results for formality transfer across languages in Table 1, Table 2. Overall we find that each of our proposed methods (DIFFUR, *-INDIC, +BT) help improve performance over the baseline UR model (71.1, 58.3, 54.2 vs 30.4 r-AGG on Hindi). Combining these ideas with multi-task learning (DIFFUR-MLT) gives us the best performance of across all languages (78.1 on Hindi). On Gujarati, the DIFFUR model fail to get good performance (0.4, 36.0 r-AGG) since they did not see Gujarati paraphrase data (Appendix A), but this performance is recovered using DIFFUR-MLT (75.0). In Table 3 we see human evaluations support our au-

Input	Generations	Analysis
Informal अपनी वाली जॉब मुझे मत बताओ. (don't tell me about your job)	*Formal* ($\lambda = 0.5$) अपनी वाली नौकरी मुझे मत बताओ। ($\lambda = 1.0$) अपनी नौकरी के बारे में मुझे बताने की जरूरत नहीं। ($\lambda = 1.5$) आपकी नियुक्ति के बारे में मुझे ना बताएं।	As sentences get more formal, the english word "job" (जॉब) is converted to Persian (नौकरी) / high Sanskrit (नियुक्ति) and honorifics are used (आपकी, बताएं)
Formal हिंसा में दो लोगों की मौत हुई थी और लगभग 150 घायल हुए थे। (two people died in the violence and 150 were injured)	*Informal* ($\lambda = 1.0$) हिंसा में दो लोग मारे गए और 150 के करीब लोग घायल हो गए. ($\lambda = 1.5$) हिंसा में 2 लोग मारे गए थे व 150 लोग घायल हुए थे ($\lambda = 2.0$) हिंसा में 2 लोग मारे गए और 150 घायल	As sentences get more informal besides lexical changes, sentence shortening is common, while roughly conveying same meaning
Positive Sentiment मुझे यह फिल्म बहुत पसंद आई तुम तोह काफी इंटेलीजेंट हो	*Negative Sentiment* इस फिल्म को मैंने कभी पसंद नहीं किया. तुम बेहद अनाड़ी हो.	Negations (नहीं) and word antonyms (इंटेलीजेंट, अनाड़ी) are common as sentiment changes
Complex भाजपा व्यंग्य करती महसूस होती है। कठिन परिश्रम कर सकता है.	*Simple* भाजपा मजाक करती दिख रही है। कड़ी चीजें कर सकते हैं।	Lexical substitutions (व्यंग्य → मजाक, कठिन → कड़ी) to use more commonly spoken words
Monocode 01.2017 से, अर्थात इस योजना के चालू होने की तिथि से प्रभावी बोली लगाने के लिए सलाहकारी सेवाएं	*Code-mixed* 01.2017, i.e. उस डेट से, जब से यह योजना इंटीग्रेटेड है बोली लगाने के लिए काउंसलिंग सर्विसिज़	With code-mixing, several english words are introduced (तिथि → डेट / date, अर्थात → i.e., सलाहकारी सेवाएं → काउंसलिंग सर्विसिज़ / counseling services)
De-anonymized फिल्म में कार्थी और अदिति राव हैदरी मुख्य किरदार निभाते हुए नजर आ रहे हैं। और इसमाईल, अलयसअ, यूनुस और लूत को भी। इनमें से हर एक को हमने संसार के मुक़ाबले में श्रेष्ठता प्रदान की	*Anonymized *** फिल्म में PII और PII PII मुख्य भूमिका निभाते हुए नजर आ रहे हैं। और PII, PII, PII और PII को भी। इनमें से प्रत्येक को हमने संसार के विरुद्ध ऊँचाइयाँ प्रदान की	Entities (अदिति राव हैदरी, इसमाईल) are replaced with PII (Personal Identifiable Information) tags, to anonymize text
Gendered रियो ओलंपिक : बैडमिंटन में भारतीय महिलाओं ने किया निराश, हार से हुई शुरुआत	*Gender Neutral *** रियो ओलंपिक : बैडमिंटन में भारतीय खिलाड़ियों ने किए निराश, हार से हुए शुरू	Gendered words (महिलाओं) are replaced with their neutral equivalents (खिलाड़ियों)

Figure 4: Outputs from our best performing model for several attribute transfer tasks (λ is style transfer magnitude). Qualitatively, we noticed lower success rates for styles marked with **; Appendix J has more model outputs.

tomatic evaluation for formality transfer. In Table 4 we perform human evaluation on a subset of models for code-mixing addition and see similar trends, with DIFFUR-MLT significantly outperforming UR, UR-INDIC (41.5 AGG vs 3.6, 15.3 on Hindi).

DIFFUR-MLT and DIFFUR-INDIC are best at controlling magnitude of style transfer: In Table 5, we compare the extent to which models can control the amount of style transfer using λ. We find that all our proposed methods outperform the UR model, which gets only 29.2 CALIB. +BT models are not as effective at control (43.4 CALIB), while DIFFUR-INDIC and DIFFUR-MLT perform best (69.6, 69.0 CALIB). This is graphically illustrated in Figure 3. DIFFUR-MLT performs consistently well across different λ values (left plot), and gives a high style change without much drop in content similarity to the input as λ is varied (right plot); more control experiments in Appendix F.

In Appendix I we provide a **breakdown by individual metrics** and plots showing variation with λ. In Appendix G we show **ablations studies** justifying the DIFFUR design, decoding scheme, etc. We also analyze the style encoder f_{style} in Appendix H, finding it is an effective style classifier.

We **analyze several qualitative outputs** from DIFFUR-MLT in Figure 4. Besides formality transfer and code-mixing addition, we transfer several other attributes: sentiment (Li et al., 2018), simplicity (Xu et al., 2015), anonymity (Anandan et al., 2012) and gender neutrality (Reddy and Knight, 2016); more outputs in Appendix J.

CONCLUSION: We present a recipe for building & evaluating controllable few-shot style transfer systems needing only 3-10 style examples at inference, useful in low-resource settings. Our methods outperform prior work in formality transfer & code-mixing for 7 languages, with promising qualitative results. Future work includes further improving systems for some attributes, and considering languages where little / no translation data is available.

Ethical Considerations

Recent work has highlighted issues of stylistic bias in text generation systems, specifically machine translation systems (Hovy et al., 2020). We acknowledge these issues, and consider style transfer and style-controlled generation technology as an opportunity to work towards fixing them (for instance, gender neutralization as presented in Section 6). Note that it is important to tread down this path carefully — In Chapter 9, Blodgett (2021) argue that style is inseparable from social meaning (as originally noted by Eckert, 2008), and humans may perceive automatically generated text very differently compared to automatic style classifiers.

Our models were trained on 32 Google Cloud TPUs. As discussed in Appendix A, the UR & UR-INDIC model take roughly 18 hours to train. The DIFFUR-* and DIFFUR-MLT models are much cheaper to train (2 hours) since we finetune the pretrained UR-* models. The Google 2020 environment report mentions,[13] "TPUs are highly efficient chips which have been specifically designed for machine learning applications". These accelerators run on Google Cloud, which is carbon neutral today, and is aiming to "run on carbon-free energy, 24/7, at all of Google's data centers by 2030" (https://cloud.google.com/sustainability).

[13] https://www.gstatic.com/gumdrop/sustainability/google-2020-environmental-report.pdf

References

Rama Kant Agnihotri. 2013. *Hindi: An essential grammar*. Routledge.

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6470–6484, Online. Association for Computational Linguistics.

Balamurugan Anandan, Chris Clifton, Wei Jiang, Mummoorthy Murugesan, Pedro Pastrana-Camacho, and Luo Si. 2012. t-plausibility: Generalizing words to desensitize text. *Transactions on Data Privacy*, 5(3):505–534.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "i am borrowing ya mixing?" an analysis of english-hindi code mixing in facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126.

Su Lin Blodgett. 2021. Sociolinguistically driven approaches for just natural language processing. *UMass Amherst Doctoral Dissertations. 2092*.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: composable transformations of Python+NumPy programs. *Github*.

Eleftheria Briakou, Sweta Agrawal, Ke Zhang, Joel Tetreault, and Marine Carpuat. 2021a. A review of human evaluation for style transfer. *arXiv preprint arXiv:2106.04747*.

Eleftheria Briakou, Di Lu, Ke Zhang, and Joel Tetreault. 2021b. Olá, bonjour, salve! XFORMAL: A benchmark for multilingual formality style transfer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3199–3216, Online. Association for Computational Linguistics.

Isaac Caswell, Julia Kreutzer, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. 2021. Quality at a glance: An audit of web-crawled multilingual datasets. *arXiv preprint arXiv:2103.12028*.

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

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Proceedings of Advances in Neural Information Processing Systems*, 32:7059–7069.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 2475–2485.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Penelope Eckert. 2008. Variation and the indexical field 1. *Journal of sociolinguistics*, 12(4).

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Xavier Garcia, Noah Constant, Mandy Guo, and Orhan Firat. 2021. Towards universality in multilingual text rewriting. *arXiv preprint arXiv:2107.14749*.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. *Proceedings of the Association for Computational Linguistics*.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*.

Junxian He, Taylor Berg-Kirkpatrick, and Graham Neubig. 2020. Learning sparse prototypes for text generation. *Advances in Neural Information Processing Systems*, 33.

George Heidorn. 2000. Intelligent writing assistance. *Handbook of natural language processing*, pages 181–207.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *Proceedings of the International Conference on Learning Representations*.

Dirk Hovy, Federico Bianchi, and Tommaso Fornaciari. 2020. “you sound just like your father” commercial machine translation systems include stylistic biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1686–1690.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Yamuna Kachru. 2006. *Hindi*, volume 12. John Benjamins Publishing.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. Gluecos: An evaluation benchmark for code-switched nlp. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585.

Elizaveta Korotkova, Agnes Luhtaru, Maksym Del, Krista Liin, Daiga Deksne, and Mark Fishel. 2019. Grammatical error correction and style transfer via zero-shot monolingual translation. *arXiv preprint arXiv:1903.11283*.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762, Online. Association for Computational Linguistics.

Ritesh Kumar. 2014. Developing politeness annotated corpus of Hindi blogs. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1275–1280, Reykjavik, Iceland. European Language Resources Association (ELRA).

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. 2021. Neural data augmentation via example extrapolation. *arXiv preprint arXiv:2102.01335*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7297–7306, Online. Association for Computational Linguistics.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating style transfer for text. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Shuyo Nakatani. 2010. Language detection library for java.

Xing Niu and Marine Carpuat. 2020. Controlling neural machine translation formality with synthetic supervision. In *Association for the Advancement of Artificial Intelligence*.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018. Multi-task neural models for translating between styles within and across languages. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1008–1021.

Richard Yuanzhe Pang. 2019. Towards actual (not operational) textual style transfer auto-evaluation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, Online.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of Empirical Methods in Natural Language Processing*, Online.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Adithya Pratapa, Antonios Anastasopoulos, Shruti Rijhwani, Aditi Chaudhary, David R. Mortensen, Graham Neubig, and Yulia Tsvetkov. 2021. Evaluating the morphosyntactic well-formedness of generated texts. *arXiv preprint arXiv:2103.16590*.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of Empirical Methods in Natural Language Processing*.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2021. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages.

Justus J Randolph. 2005. Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss' fixed-marginal multirater kappa. *Online submission*.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

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

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Parker Riley, Noah Constant, Mandy Guo, Girish Kumar, David Uthus, and Zarana Parekh. 2021. TextSETTR: Few-shot text style extraction and tunable targeted restyling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3786–3800, Online. Association for Computational Linguistics.

Bidisha Samanta, Mohit Agrawal, and Niloy Ganguly. 2021. A hierarchical vae for calibrating attributes while generating text using normalizing flow. *ACL*, page 2405–2415.

Bidisha Samanta, Sharmila Reddy, Hussain Jagirdar, Niloy Ganguly, and Soumen Chakrabarti. 2019. A deep generative model for code-switched text. *arXiv preprint arXiv:1906.08972*.

Mingyue Shang, Piji Li, Zhenxin Fu, Lidong Bing, Dongyan Zhao, Shuming Shi, and Rui Yan. 2019. Semi-supervised text style transfer: Cross projection in latent space. In *Proceedings of Empirical Methods in Natural Language Processing*.

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

Rakshith Shetty, Bernt Schiele, and Mario Fritz. 2018. A4nt: author attribute anonymity by adversarial training of neural machine translation. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1633–1650.

Eric Michael Smith, Diana Gonzalez-Rico, Emily Dinan, and Y-Lan Boureau. 2020. Controlling style in generated dialogue. *arXiv preprint arXiv:2009.10855*.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text style transfer. In *Proceedings of the International Conference on Learning Representations*.

Aleksey Tikhonov and Ivan P Yamshchikov. 2018. Sounds wilde. phonetically extended embeddings for author-stylized poetry generation. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 117–124.

Alexey Tikhonov, Viacheslav Shibaev, Aleksander Nagaev, Aigul Nugmanova, and Ivan P Yamshchikov. 2019. Style transfer for texts: Retrain, report errors, compare with rewrites. In *Proceedings of Empirical Methods in Natural Language Processing*.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. *Advances in Neural Information Processing Systems*, 32:11036–11046.

Matthijs J Warrens. 2010. Inequalities between multi-rater kappas. *Advances in data analysis and classification*, 4(4):271–286.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

Peng Xu, Jackie Chi Kit Cheung, and Yanshuai Cao. 2020. On variational learning of controllable representations for text without supervision. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10534–10543. PMLR.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of International Conference on Computational Linguistics*.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2021a. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *arXiv preprint arXiv:2105.13626*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021b. mT5: A massively multilingual pre-trained text-to-text transformer. In *Conference of the North American Chapter of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Online. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Appendices for "Few-shot Controllable Style Transfer for Low-Resource Multilinugal Settings"

A Model training & evaluation details

We compare the following models:

- UR: the Universal Rewriter (Garcia et al., 2021), which is our main baseline (Section 3);
- DIFFUR: our model with paraphrase vector differences (Section 4.2);
- UR-INDIC, DIFFUR-INDIC: Indic variants of UR and DIFFUR models (Section 4.3);
- DIFFUR-MLT: Multitask training between UR-INDIC and DIFFUR-INDIC (Section 4.4);
- + BT: models with style-controlled backtranslation at inference time (Section 4.1).

To **train** the UR-INDIC model, we use mC4 (Xue et al., 2021b) for the self-supervised objectives and Samanantar (Ramesh et al., 2021) for the supervised translation. For creating paraphrase data for training our DIFFUR models (Section 4.2), we again leverage Indic language side of Samanantar sentence pairs. Our models are implemented in JAX (Bradbury et al., 2018) using the T5X library.[14] We re-use the UR checkpoint from Garcia et al. (2021). To train the UR-INDIC model, we follow the setup in Garcia et al. (2021) and initialize the model with mT5-XL (Xue et al., 2021b), which has 3.7B parameters. We fine-tune the model for 25K steps with a batch size of 512 inputs and a learning rate of 1e-3, using the objectives in Section 3. Training was done on 32 Google Cloud TPUs which took a total of 17.5 hours. To train the DIFFUR and DIFFUR-INDIC models, we further fine-tune UR and UR-INDIC for a total of 4K steps using the objective from Section 4.2, taking 2 hours.

Evaluation Datasets: Our models are evaluated on (1) formality transfer; (2) the task of adding code-mixing in text. Since we do not have access to any formality evaluation dataset,[15] we hold out 22K sentences from Samanantar in each Indic language for validation / testing. For Swahili / Spanish, we use mC4 / WMT2018 sentences. These sets have similar number of formal / informal sentences, as marked by our formality classifiers (Section 5.2), and are transferred to the opposite formality. We re-use the hi/bn formality transfer splits for code-mixing addition, where a system must increase the amount of code-mixing (with English) in a sentence, as shown in our exemplars in Appendix D.

Seven languages with varying scripts and morphological richness are used for evaluation (`hi,es,sw,bn,kn,te,gu`). The UR model only saw translation data for `hi,es,bn`, whereas UR-INDIC sees translation data for all Indic languages (Section 4.3). To test the generalization capability of the DIFFUR, no Gujarati paraphrase training data for is used.

B More Related Work

Multilingual style transfer is mostly unexplored in prior work: a 35 paper survey by Briakou et al. (2021b) found only one work in Chinese, Russian, Latvian, Estonian, French (Shang et al., 2019; Tikhonov and Yamshchikov, 2018; Korotkova et al., 2019; Niu et al., 2018). Briakou et al. (2021b) further introduced XFORMAL, the first formality transfer *evaluation* dataset in French, Brazilian Portugese and Italian.[16] Hindi formality has been studied in linguistics, focusing on politeness (Kachru, 2006; Agnihotri, 2013; Kumar, 2014) and code-mixing (Bali et al., 2014). Due to its prevalence in India, English-Hindi code-mixing has seen work in language modeling (Pratapa et al., 2018; Samanta et al., 2019) and core NLP tasks (Khanuja et al., 2020). To the best of our knowledge, we are the first to study style *transfer* for Indic languages. A few prior works build models which can **control the degree of style transfer** using a scalar input (Wang et al., 2019; Samanta et al., 2021). However, these models are style-specific and require large unpaired style corpora during training. We adopt the inference-time control method used by Garcia et al. (2021) and notice much better controllability after our proposed fixes in Section 4.2.

C More details on the translation-specific Universal Rewriter objectives

In this section we describe the details of the supervised translation objective and the style-controlled translation objective used in the Universal Rewriter

[14] `https://github.com/google-research/google-research/tree/master/flax_models/t5x`

[15] We do not use GYAFC (Rao and Tetreault, 2018) and XFORMAL (Briakou et al., 2021b) due to reasons in footnote 4. Our dataset from Section 5.1 has already been used for classifier selection, and has machine generated sentences.

[16] We do not use this data since it does not cover Indian languages, and due to Yahoo! L6 corpus restrictions for industry researchers (confirmed via authors correspondence).

model. See Section 3 for details on the exemplar-based denoising objective.

Learning translation via direct supervision: This objective is the standard supervised translation setup, using zero vectors for style. The output language code is prepended to the input. Consider a pair of parallel sentences (x, y) in languages with codes `lx`, `ly` (prepended to the input string),

$$\bar{y} = f_{\text{ur}}(\texttt{ly} \oplus x, \mathbf{0})$$
$$\mathcal{L}_{\text{translate}} = \mathcal{L}_{\text{CE}}(\bar{y}, y)$$

The Universal Rewriter is trained on English-centric translation data from the high-resource languages in OPUS-100 (Zhang et al., 2020).

Learning style-controlled translation: This objective emulates "style-controlled translation" in a self-supervised manner, via backtranslation through English. Consider x_1 and x_2 to be two non-overlapping spans in mC4 in language `lx`,

$$x_2^{\text{en}} = f_{\text{ur}}(\texttt{en} \oplus x_2, -f_{\text{style}}(x_1))$$
$$\bar{x}_2 = f_{\text{ur}}(\texttt{lx} \oplus x_2^{\text{en}}, f_{\text{style}}(x_1))$$
$$\mathcal{L}_{\text{BT}} = \mathcal{L}_{\text{CE}}(\bar{x}_2, x_2)$$

D Choice of Exemplars

Codemixed Exemplars

1. गुड मॉर्निंग, भारत
2. अगर आप इसे फ्रीज करना चाहते हैं, तो आपको टेंपेरेचर कम करना चाहिए
3. हाय मुझे जॉब चाहिए
4. हॉलीवुड एक्ट्रेस एंजेलिना जॉली एक एनिमेशन फिल्म प्रोड्यूस कर रही हैं।
5. इस टूर्नमेंट में 6 टीमें टाइटल के लिए कम्पीट् करेंगी।

Monocode Exemplars

1. सुप्रभात, भारत
2. अगर आप इसे जमाना चाहते हैं, तो आपको तापमान कम करना चाहिए
3. नमस्ते मुझे नौकरी चाहिए
4. हॉलीवुड अभिनेत्री एंजेलिना जोली एक चलचित्र का निर्माण कर रही हैं।
5. इस खेल प्रतियोगिता में छह समूह खिताब के लिए प्रतिस्पर्धा करेंगे।

Figure 5: Exemplars used for adding code-mixing.

Gendered Exemplars

1. नर्स साफ कपड़े पहनी थी
2. हमें और जनशक्ति की जरूरत है
3. यह डॉक्टर बहुत अच्छा है

Gender-neutral Exemplars

1. नर्स ने साफ कपड़े पहने थे
2. हमें और कर्मचारियों की जरूरत है
3. यह डॉक्टर बहुत अच्छे हैं

Figure 6: Exemplars used for gender neutralization.

Formal exemplars
1. This was a remarkably thought-provoking read.
2. It is certainly amongst my favorites.
3. We humbly request your presence at our gala in the coming week.

Informal exemplars
1. reading this rly makes u think
2. Its def one of my favs
3. come swing by our bbq next week if ya can make it

De-anonymized Exemplars

1. मेरा फोन नंबर 091898807646 है
2. केट का आधार नंबर है 4098-7980-8098
3. 18 सितंबर को मैंने microsoft.com पर विज़िट किया और IP 192.168.0.1 से test@google.site पर एक ईमेल भेजा।
4. मेरा पासपोर्ट नंबर 4903-3289-2394 है
5. फिल Google में बारबरा की टीम में काम करता है
6. बॉब 42 साल का है
7. शर्लक 221B बेकर स्ट्रीट में रहता है
8. मेरा ईमेल पता है email1@gmail.com

Anonymized Exemplars

1. मेरा फोन नंबर PII है
2. PII का आधार नंबर है PII
3. PII को मैंने PII पर विज़िट किया और IP PII से PII पर एक ईमेल भेजा।
4. मेरा पासपोर्ट नंबर PII है
5. PII PII में PII की टीम में काम करता है
6. PII PII साल का है
7. PII PII में रहता है
8. मेरा ईमेल पता है PII

Figure 7: Exemplars used for text anonymization. All entities in the deanonymized exemplars are random.

Complex exemplars
1. The static charges remain on an object until they either bleed off to ground or are quickly neutralized by a discharge.
2. It is particularly famous for the cultivation of kiwifruit.
3. Notably absent from the city are fortifications and military structures.
Simple exemplars
1. Static charges last until they are grounded or discharged.
2. This area is known for growing kiwifruit.
3. Some things important missing from the city are protective buildings and military buildings.

Positive sentiment exemplars
1. The most comfortable bed I've ever slept on, I highly recommend it.
2. I loved it.
3. The movie was fantastic.
Negative sentiment exemplars
1. The most uncomfortable bed I've ever slept on, I would never recommend it.
2. I hated it.
3. The movie was awful.

E Evaluation Appendix

E.1 Multilingual Classifier Selection

Due to the absence of a style classification dataset in Indic languages, we built our multilingual classifier drawing inspiration from recent research in zero-shot cross-lingual transfer (Conneau et al., 2018; Conneau and Lample, 2019; Pfeiffer et al., 2020b). We experimented with three zero-shot transfer techniques while selecting our classifiers for evaluating multilingual style transfer.

TRANSLATE TRAIN: The first technique uses the hypothesis that style is preserved across translation. We classify the style of English sentences in the Samanantar translation dataset (Ramesh et al., 2021) using a style classifier trained on English formality data from Krishna et al. (2020). We use the human translated Indic languages sentences as training data. This training data is used to fine-tune a large-scale multilingual language model.

ZERO-SHOT: The second technique fine-tunes large-scale multilingual language models on a English style transfer dataset, and applies it zero-shot on multilingual data during inference.

MAD-X: Introduced by Pfeiffer et al. (2020b), this technique is similar to ZERO-SHOT but additionally uses language-specific parameters ("adapters") during inference. These language-specific adapters have been originally trained using masked language modeling on the desired language data.

Dataset for evaluating classifiers: We conduct our experiments on Hindi formality classification, leveraging our evaluation datasets from Section 5.1. We removed pairs which did not have full agreement across the three annotators and those pairs which had the consensus rating of "Equal" formality. This filtering process leaves us with 316 pairs in Hindi (out of 1000). In our experiments, we check whether the classifiers give a higher score to the more formal sentence in the pair.

Models: We leverage the multilingual classifiers open-sourced[17] by Krishna et al. (2020). These models have been trained on the English GYAFC formality classification dataset (Rao and Tetreault, 2018), and have been shown to be effective on the XFORMAL dataset (Briakou et al., 2021b) for formality classification in Italian, French and Brazilian Portuguese.[13] These classifiers were trained on preprocessed data which had trailing punctuation stripped and English sentences lower-cased, encouraging the models to focus on lexical and syntactic choices. As base multilingual language models, we use (1) mBERT-base from Devlin et al. (2019); (2) XLM-RoBERTa-base from Conneau et al. (2020).

Results: Our results on Hindi are presented in Table 6 and other languages in Table 7. Consistent with Pfeiffer et al. (2020b), we find MAD-X to be a superior zero-shot cross lingual transfer method compared to baselines. We also find XLM-R has better multilingual representations than mBERT. Unfortunately, AdapterHub (Pfeiffer et al., 2020a) has XLM-R language adapters available only for Hindi & Tamil (among Indic languages). For other languages we use the ZERO-SHOT technique on XLM-R, consistent with the recommendations[13] provided by Krishna et al. (2020) based on their ex-

[17] https://github.com/martiansideofthemoon/style-transfer-paraphrase/blob/master/README-multilingual.md

periments on XFORMAL (Briakou et al., 2021b).

Method	Model	Accuracy (↑)
TRANSLATE TRAIN	mBERT	66%
ZERO-SHOT	mBERT	72%
	XLM-R	76%
MAD-X	XLM-R	**81%**

Table 6: Hindi formality classification accuracy on our crowdsourced dataset (Section 5.1) using different cross-lingual transfer methods. Our results indicate that MAD-X is the most effective method, and XLM-R is a better pretrained model than mBERT.

Language	mBERT	XLM-R
bn	65.3%	82.2%
kn	76.3%	76.9%
te	72.6%	74.6%

Table 7: Formality classification on our crowdsourced Bengali, Kannada and Telugu dataset (Section 5.1) using the ZERO-SHOT technique described in Appendix E.1. Results confirm the efficacy of the XLM-R classifier. See Table 6 for Hindi results.

E.2 Semantic Similarity Model Selection

We considered three models for evaluating semantic similarity between the input and output:

(1) LaBSE (Feng et al., 2020);
(2) m-USE (Yang et al., 2020);
(3) multilingual Sentence-BERT (Reimers and Gurevych, 2020), the knowledge-distilled variant `paraphrase-xlm-r-multilingual-v1`

Among these models, only LaBSE has support for all the Indic languages we were interested in. No Indic language is supported by m-USE, and multilingual Sentence-BERT has been trained on parallel data only for Hindi, Gujarati and Marathi among our Indic languages. However, in terms of Semantic Textual Similarity (STS) benchmarks (Cer et al., 2017) for English, Arabic & Spanish, m-USE and Sentence-BERT outperform LaBSE (Table 1 in Reimers and Gurevych, 2020).

LaBSE correlates better than Sentence-BERT with our human-annotated formality dataset: We measured the Spearman's rank correlation between the semantic similarity annotations on our human-annotated formality datasets (Section 5.1). We discarded 10% sentence pairs which had no agreement among three annotators and took the majority vote for the other sentence pairs. We assigned "Different Meaning" a score of 0, "Slight Difference in Meaning" a score of 1 and "Approximately Same Meaning" a score of 2 before measuring Spearman's rank correlation. In Table 8 we see a stronger correlation of human annotations with LaBSE compared to Sentence-BERT, especially for languages like Bengali, Kannada for which Sentence-BERT did not see parallel data.

Model	hi	bn	kn	te
LaBSE	0.34	0.49	0.39	0.25
Sentence-BERT	0.33	0.36	0.29	0.18

Table 8: Spearman's rank correlation between different semantic similarity models and our semantic similarity human annotations collected along with formality labels. Overall, LaBSE correlates more strongly than Sentence-BERT with our annotated data.

E.3 Evaluation with Different LaBSE thresholds

In Section 6, we set our LaBSE threshold L to 0.75. In this section, we present our evaluations with a more and less conservative value of L.

In Table 17, we present results with $L = 0.65$, and in Table 18 we set $L = 0.85$. Compared to Table 1, trends are mostly similar, with DIFFUR models and INDIC variants outperforming counterparts. Note that the absolute values of SIM and AGG metrics differ, with absolute values going down with the stricter threshold of $L = 0.85$, and up with the relaxed threshold of $L = 0.65$.

Comparing chosen thresholds with human annotations: To verify these three thresholds are reasonable choices, we measure the LaBSE similarity of the sentence pairs annotated by humans, and compare the LaBSE scores to human semantic similarity annotations. We pool the "Approximately Same Meaning" and "Slight Difference in Meaning" categories as "same", and consider only sentence pairs with a majority rating of "same". In Table 9 we see that the chosen thresholds span the spectrum of LaBSE values for the human annotated semantically similar pairs.

Threshold L	% of sentence pairs > L hi	bn	kn	te
0.65	97.4	96.1	94.6	90.6
0.75	83.9	76.1	68.4	62.6
0.85	75.1	62.7	50.5	45.5

Table 9: Percentage of human annotated semantically similar pairs which have a LaBSE score of at least L. As we increase the threshold L, we see this percentage substantially reduces, indicating our chosen thresholds are within the range of variation in LaBSE scores for semantically similar sentences.

E.4 More Crowdsourcing Details

In Figure 16, we show screenshots of our crowdsourcing interface along with all the instructions shown to crowdworkers. The instructions were written after consulting professional Indian linguists. Each crowdworker was allowed to annotate a maximum of 50 different sentence pairs per language, paying them \$0.05 per pair. For formality classification, we showed crowdworkers two sentences and asked them to choose which one is more formal. Crowdworkers were allowed to mark ties using an "*Equal*" option. For semantic similarity annotation, we showed crowdworkers the sentence pair and provided three options — "*approximately same meaning*", "*slight difference in meaning*", "*different meaning*", to emulate a 3-point Likert scale. While performing our human evaluation (Section 5.7), we use a 0.5 SIM score for "*slight difference in meaning*" and a 1.0 SIM score for "*approximately same meaning*" annotations. For every system considered, we analyzed the same set of 200 input sentences for style transfer performance, and 100 of those sentences for evaluating controllability. We removed sentences which were exact copies of the input (after removing trailing punctuation) or were in the wrong language to save annotator time and cost. When outputs were exact copies of the input, we assigned SIM = 100, ACC = 0, AGG = 0.

In Table 10 and Table 11 we show the inter-annotator agreement statistics. We measure Fleiss Kappa (Fleiss, 1971), Randolph Kappa (Randolph, 2005; Warrens, 2010), the fraction of sentence pairs with total agreement between the three annotators and the fraction of sentence pairs with no agreement.[18] In the table we can see all agreement statistics are well away from a uniform random annotation baseline, indicating good agreement.

[18]The κ scores are measured using the library https://github.com/statsmodels/statsmodels.

	F-κ	R-κ	all agree	none agree
Random	0.0	0.0	11.1%	22.2%
hi	0.21	0.28	32.8%	10.2%
bn	0.33	0.40	43.8%	7.2%
kn	0.22	0.31	35.0%	7.7%
te	0.21	0.31	36.0%	9.3%

Table 10: Fleiss kappa (F-κ), Randolph kappa (R-κ), and agreement scores of crowdsourcing for **formality classification**. All κ scores are well above a random annotation baseline, indicating fair agreement.

	F-κ	R-κ	all agree	none agree
Random	0.0	0.0	11.1%	22.2%
hi	0.10	0.27	32.6%	11.8%
bn	0.24	0.34	38.7%	10.2%
kn	0.13	0.25	30.8%	11.3%
te	0.1	0.31	36.1%	9.7%

Table 11: Fleiss kappa (F-κ), Randolph kappa (R-κ), and agreement scores of crowdsourcing for **semantic similarity**. All κ scores are well above a random annotation baseline, indicating fair agreement.

E.5 Fluency Evaluation

Unlike some prior works, we **avoid evaluation of output fluency** due to the following reasons: (1) lack of fluency evaluation tools for Indic languages;[19] (2) fluency evaluation often discriminates against styles which are out-of-distribution for the fluency classifier, as discussed in Appendix A.8 of Krishna et al. (2020); (3) several prior works (Pang, 2019; Mir et al., 2019; Krishna et al., 2020) have recommended against using perplexity of style language models for fluency evaluation since it is unbounded and favours unnatural sentences with common words; (4) large language models are known to produce fluent text as perceived by humans (Ippolito et al., 2020; Akoury et al., 2020), reducing the need for this evaluation.

[19]A potential tool for fluency evaluation in future work is LAMBRE (Pratapa et al., 2021). However, the original paper does not evaluate performance on Indic languages and the grammars for Indic languages would need to collected / built.

E.6 Details of other individual metrics

Language Consistency (LANG): Since our semantic similarity metric LaBSE is language-agnostic, it tends to ignore accidental translations, which are common errors in large multilingual transformers (Xue et al., 2021a,b), especially the Universal Rewriter (Section 3.1). Hence, we check whether the output sentence is in the same language as the input, using `langdetect`.[20]

Output Diversity (COPY, 1-g): As discussed in Section 3.1, the Universal Rewriter has a strong tendency to copy the input verbatim. We build two metrics to measure output diversity compared to the input, which have been previously used for extractive question answering evaluation (Rajpurkar et al., 2016). The first metric COPY measures the fraction of outputs which were copied verbatim from the input. This is done after removing trailing punctuation, to penalize models generations which solely modify punctuation. A second metric 1-g measures the unigram overlap F1 score between the input and output. A diverse style transfer system should minimize both COPY and 1-g.

F More Controllability Evaluations

We follow the setup in Section 5.6 to first compute a λ_{max} per system. We then compute the following,

1. **Style Transfer Performance** (r-AGG): An ideal system should have good overall performance (Section 5.5) across different values in the range Λ.
2. **Average Style Score Increase** (INCR): As our control value increases, we want the classifier's target style score (compared to the input) to increase. Additionally, we want the style score increase of λ_{max} to be as high as possible, indicating the system can span the range of classifier scores.
3. **Style Calibration to λ** (CALIB, C-IN): As defined in Section 5.6. We additionally also measure calibration by including the input sentence x in the CALIB(x) calculation, treating it as the output for $\lambda = 0$ (no style transfer). Here, calibration is averaged over a total of $n = 6$ (λ_1, λ_2) pairs. We call this metric C-IN.

A detailed breakdown of performance by different metrics for every model is shown in Table 14.

[20] This package is the Python port of Nakatani (2010).

G Ablation Studies

G.1 Ablation Study for DIFFUR design

This section describes the ablation experiments conducted for the DIFFUR modeling choices in Section 4.2. We ablate a DIFFUR-INDIC model trained on Hindi paraphrase data only, and present results for Hindi formality transfer in Table 15.

- **no paraphrase**: We replaced the paraphrase noise function with the random token dropping / replacing noise used in the denoising objective of UR model (Section 3), and continued to use vector differences. As seen in Table 15, this significantly increases the copy rate, which lowers the style transfer performance.

- **no paraphrase semantic filtering**: We keep a setup identical to Section 4.2, but avoid the LaBSE filtering done (discarding pairs having a LaBSE score outside [0.7, 0.98]) to remove noisy paraphrases or exact copies. As seen in Table 15, this decreases the semantic similarity score of the generations, lowering the overall performance.

- **no vector differences**: Instead of using vector differences for DIFFUR-INDIC, we simply set $\mathbf{s}_{\text{diff}} = f_{\text{style}}(x)$, or the style of the target sentence. In Table 15, we see this significantly decreases SIM scores, and LANG scores for $\lambda = 2.0$. We hypothesize that this training encourages the model to rely more heavily on the style vectors, ignoring the paraphrase input. This could happen since the style vectors are solely constructed from the output sentence itself, and semantic information / confounding style is not subtracted out. In other words, the model is behaving more like an autoencoder (through the style vector) instead of a denoising autoencoder with stylistic supervision.

- **mC4 instead of Samanantar**: Instead of creating pseudo-parallel data with Samanantar, we leverage the mC4 dataset itself which was used to train the UR model. We backtranslate spans of text from the Hindi split of mC4 on-the-fly using the UR translation capabilities, and use it as the "paraphrase noise function". To ensure translation performance does not deteriorate during training, 50% minibatches are supervised translation between Hindi and English. In Table 15, we see decent overall performance, but the LANG score is 6% lower than DIFFUR-INDIC. Qualitatively we found that the model often translates a few Hindi words to English while making text informal. Due to sparsity of English tokens, it often escapes penalization from LANG.

- **mC4 + exemplar instead of target**: This setting is similar to the previous one, but in addition to the mC4 dataset we utilize the vector difference between the style vector of the exemplar span (instead of target span), and the "paraphrase noised" input. Results in Table 15 show this method is not effective, and it's important for the vector difference to model the precise transformation needed.

G.2 Choice of Decoding Scheme

We experiment with five decoding schemes on the Hindi formality validation set — beam search with beam size 1, 4 and top-p sampling (Holtzman et al., 2020) with $p = 0.6, 0.75, 0.9$.

In Table 16, we present results at a constant style transfer magnitude ($\lambda = 3.0$). Consistent with Krishna et al. (2020), we find that top-p decoding usually gets higher style accuracy (r-ACC, a-ACC) and output diversity (1-g, COPY) scores, but lower semantic similarity (SIM) scores. Overall beam search triumphs since the loss in semantic similarity leads to a worse performing model. In Figure 9, we see a consistent trend across different magnitudes of style transfer (λ). In all our main experiments, we use beam search with beam size 4 to obtain our generations.

G.3 Number of Training Steps

In Figure 10, we present the variation in style transfer performance with number of training steps for our best model, the DIFFUR-MLT model. We find that with more training steps performance generally improves, but improvements saturate after 8k steps. We also see the peak of the graphs (best style transfer performance) shift rightwards, indicating a preference for higher λ values.

H Analysis Experiments

H.1 Style vectors from f_{style} as style classifiers

The Universal Rewriter models succeed in learning an effective style space, useful for few-shot style transfer. But can this metric space also act as a style classifier? To explore this, we measure the cosine distance between the mean style vector of our informal exemplars,[21] and the style vectors derived by passing human-annotated formal/informal pairs (from our dataset of Section 5.1) through f_{style}. We only consider pairs which had complete agreement among annotators. In Table 12 we see good agreement (68.2%-80.7%) between human annotations and the classifier derived from the metric space of the UR-INDIC model. Agreement is lower (67.0%-74.3%) for the DIFFUR-INDIC model, likely due to the stop gradient used in Section 4.2. With DIFFUR-MLT, agreement jumps back up to 75%-81.7% since gradients flow into the style extractor as well.

Model	hi	bn	kn	te
UR	79.1	69.7	66.2	67.1
UR-INDIC	80.7	74.3	68.2	72.2
DIFFUR-INDIC	68.0	73.8	67.0	70.4
DIFFUR-MLT	75.0	81.7	79.8	79.0

Table 12: style vector as a classifier, measuring the cosine similarity with informal exemplar vectors.

H.2 Style Vector Analysis with Formal Exemplars Vectors

In Appendix H.1, we saw that the metric vector space derived from the style encoder f_{style} of various models is an effective style classifier, using the *informal* exemplar vectors. In Table 13, we present a corresponding analysis using *formal* exemplar vectors. Most accuracy scores are close to 50%, implying this setup is not a very effective style classifier.

Model	hi	bn	kn	te
UR	56.6	60.0	61.6	57.6
UR-INDIC	59.5	60.6	52.6	44.8
DIFFUR-INDIC	58.5	58.3	59.5	49.7
DIFFUR-MLT	64.9	52.3	47.1	41.8

Table 13: style vector as a classifier, measuring the cosine similarity with formal exemplar vectors.

I Full Breakdown of Results

A full breakdown of results by individual metrics, along with plots showing variation with change in

[21] See Appendix D for the exemplar sentences. We found the informal exemplars more effective than formal exemplars for style classification; Appendix H.2 has a comparison.

λ, is provided for — Hindi (Table 19, Figure 11), Bengali (Table 20, Figure 12), Kannada (Table 21, Figure 13), Telugu (Table 22, Figure 14), Gujarati (Table 23, Figure 15).

In the baseline Hindi UR model, we notice high COPY rates (45.4%), resulting in lower ACC scores. COPY reduces in our proposed models (4.4% for DIFFUR-MLT), which boosts overall performance. We find the lowest COPY (and lowest 1-g) for models with +BT (1%), which is due to two steps of translation. However, this lowers semantic similarity (also seen in Table 3) lowering the overall score compared to DIFFUR-MLT (60.0 vs 78.1 r-AGG).

J More Model Outputs

Please refer to Figure 8. In the main body, Figure 4 has a few examples as well with detailed analysis.

Input	Generations	Input English Translation
Complex	*Simple*	
कोर्ट के आदेशों की अनदेखी	कोर्ट के बातों को नजर अंदाज	They ignored the court orders
श्री मोदी हिन्दी बोलने वाले प्रधानमंत्री हैं और उन्होंने देश-विदेश में हिन्दी का मान बढ़ाया है।	श्री मोदी हिन्दी बोलने वाले पीएम जिन्होंने देश और विदेश में हिन्दी बढ़ाई है।	Narendra Modi is a Hindi speaking prime minister who has popularized Hindi across the world
पुलिस ने दिल्ली से पांच लोगों को गिरफ्तार किया है।	दिल्ली से पांच लोगों को पकड़ा	The police arrested 5 people in Delhi
Simple	*Complex*	
वह बॉम्बे हाईकोर्ट के सबसे सीनियर जज हैं।	वह बॉम्बे हाईकोर्ट के सर्वाधिकृत न्यायाधीश हैं	He/She is the most senior judge in the Bombay High Court.
मैंने उनके साथ बहुत करीब से काम किया है।	मैंने उनके साथ बहुत निकटता से काम किया है।	I've worked closely with them.
Informal	*Formal*	
फिल्म इंडस्ट्री में करती है काम	वह फिल्मी जगत में महत्वपूर्ण भूमिका निभाती हैं	He/She plays an important role in the film industry.
अरे भई, हम कोई मज़ाक़ नहीं कर रहे.	प्रियजनों, हम कोई हँसी-खेल नहीं कर रहे हैं.	Friends, this is not a joke.
तुम जियो या मरो मुझे इससे कोई मतलब नहीं है.	आप जीते या मरते हैं, इससे मुझे कोई मतलब नहीं है.	I don't care whether you live or die!
उसके बाद तो जैसे बवाल मच गया.	बाद में जैसे-जैसे हड़कंप मच गया	After this there was a lot of chaos
और जोश व ख़रोश वाले समन्दर की	और जोश और आशान्वित समूह की	In the sea of energy and passion
Formal	*Informal*	
अभिभावक भी अपनी लड़कियों को इन महाविद्यालयों में प्रवेश दिलवाने के इच्छुक हैं।	अभिभावक भी अपनी लड़कियों को इन कॉलेजों में भेजने के इच्छुक हैं	Parents also wish to get their daughters admitted in these colleges.
दूसरों की बात प्यार से सुनने में यीशु मसीह एक बेहतरीन मिसाल है।	दूसरे की बात सुनने में यीशु मसीह बेस्ट है	Jesus Christ is the best example of an empathetic listener.
Positive Sentiment	*Negative Sentiment*	
यह होटल काफी अच्छा था	यह होटल बहुत बुरा था.	This hotel was very good.
Negative Sentiment	*Positive Sentiment*	
पता नहीं चलता, लेकिन फिल्म के प्रति बेरूखी बढ़ती जाती है	पता नहीं, लेकिन फिल्म के प्रति दर्शकों की रुचि बढ़ती जा रही है	You don't realize, but your interest towards the film continually declines as you watch it
कार्यालय के कर्मचारी और प्रशासन बहुत खराब है	कार्यालय के कर्मचारी और प्रशासनिक प्रबंधन बहुत अच्छे हैं	Office staff and administrative management are very good
Monocode	*Code-mixed*	
यहां कोई मूलभूत सुविधाएं नहीं हैं।	यहां कोई बुनियादी फीचर्स नहीं हैं।	This doesn't even have basic features.
झपटमारी में शामिल एक व्यक्ति को पकड़ा।	गिरोह के एक शख्स को रिमांड पर लिया	One person involved in the prank was caught.
इन 11 अभियुक्तों में से किसी के नाम की जानकारी नहीं दी गई है.	इन 11 आरोपियों में से किसी का नाम लीक नहीं किया गया है।	The names of the 11 accused have not been revealed.
यह बारिश कई प्रदेशों में हुई है.	यह लॉकडाउन राज्य के कई हिस्सों में हुआ है।	It rained in several states.
शिवसेना और बीजेपी में कोई अंतर नहीं है	शिवसेना और बीजेपी में कोई गुड न्यूज नहीं है।	There's no difference between Shiv Sena and the BJP.
De-anonymized	*Anonymized*	
2019 लोकसभा चुनाव के लिए प्रशांत किशोर ने शुरू किया काम	2019 लोकसभा चुनाव के लिए PII ने शुरू किया काम	Prashant Kishore has started working for the 2019 Lok Sabha elections
इसके बाद आकर इंदिरा गांधी ने स्वर्ण मंदिर पर हमला किया	इसके बाद PII ने PII पर हमला किया	After this, Indira Gandhi ordered an attack on the Golden Temple
निरंजन एक नर्तकी, मल्लिका और अमीरचंद द्वारा गुमराह किया जाता है, जो उसके धन के बाद हैं।	निरंजन को एक PII, PII और PII द्वारा गुमराह किया जाता है, जो उसके धन के बाद हैं।	Niranjan is misled by a dancer, Mallika & Amirchand, who are after his wealth.

Figure 8: More qualitative examples of generations from our system (see Figure 4 for main table with qualitative analysis). Red and blue colours indicate attribute-specific features, while golden text represents model errors.

Model	$\lambda_{max}/3$			$2\lambda_{max}/3$			λ_{max}			Overall	
	λ	r-AGG	INCR	λ	r-AGG	INCR	λ	r-AGG	INCR	CALIB	C-IN
UR (2021)	0.5	22.1	5.2	1.0	26.9	8.9	1.5	30.4	18.7	29.2	31.6
UR-INDIC	0.5	53.2	13.4	1.0	58.3	18.8	1.5	54.6	26.7	60.7	65.1
UR + BT	0.3	53.2	21.4	0.7	53.9	23.5	1.0	49.1	26.9	43.4	58.8
UR-INDIC + BT	0.3	57.3	22.9	0.7	59.4	24.6	1.0	60.0	26.7	38.7	56.0
DIFFUR	0.5	65.8	16.6	1.0	71.1	26.0	1.5	67.1	21.9	64.9	72.5
DIFFUR-INDIC	0.8	67.2	17.9	1.7	72.6	27.3	2.5	65.0	**36.7**	**69.6**	**75.5**
DIFFUR-MLT	0.8	56.6	11.3	1.7	72.6	18.1	2.5	78.1	29.9	69.0	71.8

Table 14: Evaluation of extent to which the magnitude of hindi formality transfer can be controlled with λ. We find that DIFFUR-INDIC, DIFFUR-MLT are best at calibrating style change to input λ (CALIB, C-IN), giving the higher style score increase (INCR) at $\lambda = \lambda_{max}$ (details of evaluation setup and metrics in Section 5.6, Appendix F).

Ablation	COPY(↓)	LANG	SIM	r-ACC	a-ACC	r-AGG	a-AGG
DIFFUR-INDIC (hindi only)	2.0	97.0	78.4	89.8	39.7	67.3	24.6
- no paraphrase**	21.0	98.3	92.2	60.0	15.7	51.9	10.7
- no paraphrase ($p, \lambda = 0.6, 3$)	14.2	98.7	81.0	70.9	28.1	51.6	12.5
- no paraphrase semantic filtering	2.2	97.2	72.2	89.1	38.6	60.7	19.6
- no vector differences**	0.0	54.3	3.2	99.0	90.0	2.4	1.0
- no vector differences ($\lambda = 0.5$)	0.9	97.4	66.8	86.4	36.5	53.5	17.3
- mC4 instead of Samanantar	1.5	91.4	82.0	89.3	39.0	67.7	24.2
- mC4 + exemplar instead of target	5.5	23.8	82.3	77.2	32.3	13.8	3.2

Table 15: Ablation study on Hindi formality transfer validation set using beam size of 4 and $\lambda = 2.0$ unless the optimal hyperparameters were different (marked by **). As shown by the overall a-AGG scores, removing any component of our design leads to an overall performance drop, sometimes significantly. For a detailed description of analysis and results, see Appendix G.1. For detailed metric descriptions, see Section 5.

Decoding	COPY(↓)	1-g(↓)	LANG	SIM	r-ACC	a-ACC	r-AGG	a-AGG
beam 4	1.8	52.7	95.8	73.3	94.7	51.6	**66.2**	**32.3**
beam 1	1.2	47.4	92.3	61.7	95.7	62.5	55.8	31.4
top-p 0.6	1.0	45.3	91.5	56.6	96.2	65.9	51.3	29.9
top-p 0.75	0.9	43.1	90.3	52.4	96.3	69.0	47.3	28.2
top-p 0.9	0.7	40.4	89.4	46.8	96.6	71.7	42.4	26.5

Table 16: Automatic evaluation of different decoding algorithms (top-p sampling and beam search) on the DIFFUR-MLT model for Hindi formality transfer (validation set) using $\lambda = 3.0$. As expected, output diversity (1-g, COPY) and style accuracy (r-ACC, a-ACC) improves as we move down the table, but compromise semantic preservation (SIM), bringing the overall performance (r-AGG, a-AGG) down. Also see Figure 9 for a comparison across λ values, and Section 5 for detailed metric descriptions.

Model	Hindi		Bengali		Kannada		Telugu		Gujarati	
	r-AGG	a-AGG	r-AGG	a-AGG	r-AGG	a-AGG	r-AGG	a-AGG	r-AGG	a-AGG
UR (2021)	34.5	13.4	33.8	9.0	26.8	8.8	24.3	10.7	25.6	5.9
UR + BT	61.6	24.2	65.6	22.8	48.8	16.0	48.7	17.6	56.3	15.1
DIFFUR	79.4	30.3	81.7	36.0	79.0	43.4	79.7	38.0	0.5	0.2
UR-INDIC	62.0	23.9	69.3	29.3	64.6	22.2	65.0	25.8	59.0	13.8
UR-INDIC + BT	68.0	28.1	73.5	33.3	72.6	29.7	71.6	31.4	68.4	21.7
DIFFUR-INDIC	80.0	32.4	80.0	32.3	79.9	41.4	78.8	37.0	38.9	16.2
DIFFUR-MLT	85.8	45.2	86.0	48.3	86.9	54.4	86.1	51.7	78.8	41.3

Table 17: Test set performance across languages for a **smaller LaBSE semantic similarity threshold** of 0.65. Due to the more relaxed threshold, absolute numbers compared to Table 1 are higher. Trends remain similar, with the DIFFUR and INDIC variants outperforming other competing methods.

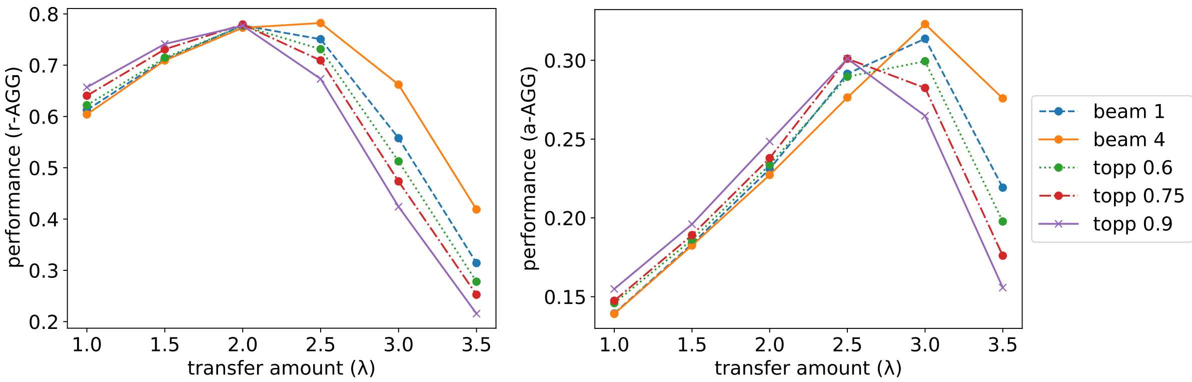

Figure 9: Variation in Hindi formality transfer (validation set) performance vs λ with change in decoding scheme, for the DIFFUR-MLT model. The plots show overall style transfer performance, using the r-AGG (left) and a-AGG (right) metrics from Section 5.5. Beam search with beam size 4 performs best, see Table 16 for an individual metric breakdown while keeping $\lambda = 3.0$.

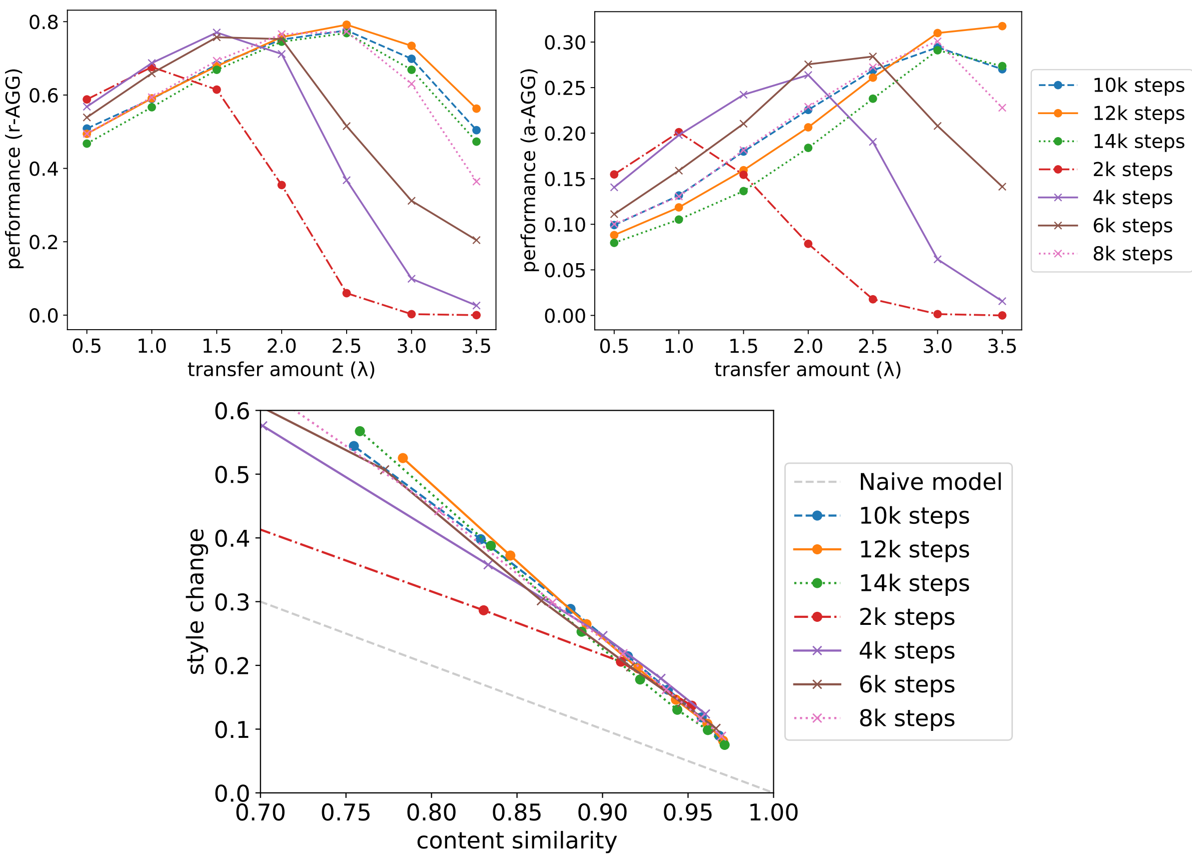

Figure 10: Variation in Hindi formality transfer validation set performance with change in number of training steps for the DIFFUR-MLT model. The plots show overall style transfer performance, using the r-AGG (top-left) and a-AGG (top-right) metrics from Section 5.5. With more training steps performance seems to improve and the peak of the graph shifts towards the right (a preference towards higher scale values). We also see more training steps leads to better controllability (bottom plot, closer to Y-axis is better), but only marginal gains after 6k steps.

Model	Hindi		Bengali		Kannada		Telugu		Gujarati	
	r-AGG	a-AGG	r-AGG	a-AGG	r-AGG	a-AGG	r-AGG	a-AGG	r-AGG	a-AGG
UR (2021)	24.2	6.6	24.2	4.8	21.5	6.0	19.1	5.8	19.4	3.6
UR + BT	40.0	10.7	31.7	8.1	21.2	5.1	19.1	4.8	26.1	4.4
DIFFUR	57.1	13.0	59.6	13.0	54.5	13.8	52.8	12.8	0.2	0.0
UR-INDIC	49.6	13.1	54.6	12.7	50.0	11.4	48.1	11.2	45.9	6.8
UR-INDIC + BT	43.7	12.9	33.9	10.2	31.9	7.8	29.4	7.8	34.0	7.4
DIFFUR-INDIC	59.2	14.9	63.8	15.6	58.9	16.1	55.2	14.4	31.7	8.0
DIFFUR-MLT	64.8	17.9	69.8	22.0	69.3	23.5	67.5	20.6	64.0	18.2

Table 18: Test set performance across languages for a **larger LaBSE semantic similarity threshold** of 0.85. Due to the stricter threshold, absolute numbers compared to Table 1 are lower, however trends are similar, with the DIFFUR and INDIC variants outperforming other competing methods.

Model	λ	COPY(↓)	1-g(↓)	LANG	SIM	r-ACC	a-ACC	r-AGG	a-AGG
UR (Garcia et al., 2021)	1.5	45.4	77.5	98.0	84.8	45.8	22.9	30.4	10.4
UR-INDIC	1.0	10.4	70.7	95.0	93.8	67.2	23.3	58.3	18.6
UR + BT	0.5	0.8	44.2	92.9	85.2	72.3	27.8	54.2	17.8
UR-INDIC + BT	1.0	1.1	49.5	95.9	85.1	76.3	33.1	60.0	22.2
DIFFUR	1.0	4.7	61.6	97.7	89.7	82.4	31.0	71.1	22.9
DIFFUR-INDIC	1.5	5.3	63.7	98.0	91.9	81.6	30.5	72.5	23.7
	2.0	3.4	57.5	98.3	84.8	86.4	36.8	70.6	24.0
DIFFUR-MLT	2.5	4.4	61.9	97.2	89.7	89.7	34.0	78.1	27.5
	3.0	2.0	52.5	95.9	72.1	94.1	51.9	64.8	32.2

Table 19: Performance breakdown of Hindi formality transfer by individual metrics described in Section 5.

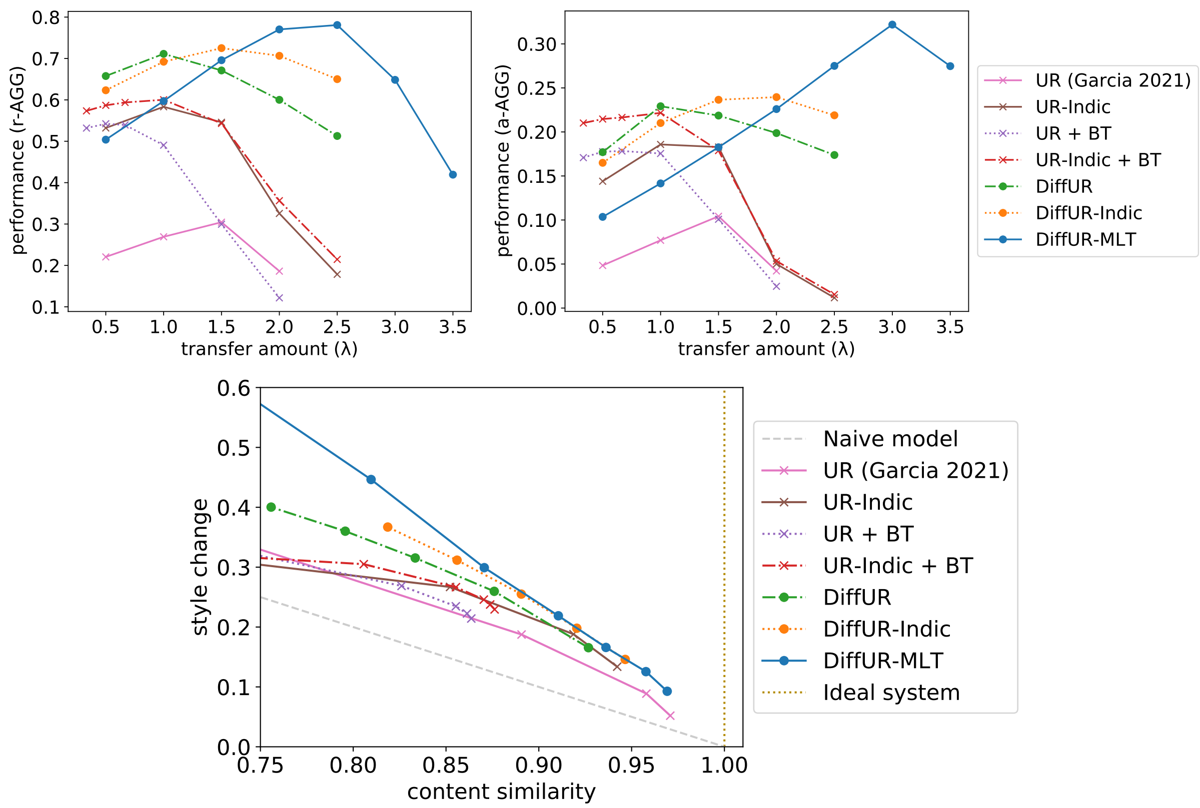

Figure 11: Variation in **Hindi** formality transfer test set performance & control for **different models** (see Table 19 for a individual metric breakdown of the models at the best performing λ). The plots show overall style transfer performance, using the r-AGG (top-left) and a-AGG (top-right) metrics from Section 5.5. We see the DIFFUR models outperform other systems across the λ range, and get best performance with the DIFFUR-MLT variant. We also see that DIFFUR models, especially with DIFFUR-MLT, lead to better style transfer control (bottom plot, closer to $x = 1$ is better), giving large style variation with λ without loss in semantics (X-axis).

Model	λ	COPY(↓)	1-g(↓)	LANG	SIM	r-ACC	a-ACC	r-AGG	a-AGG
UR (Garcia et al., 2021)	1.5	21.5	69.1	99.9	87.3	42.4	15.6	30.4	7.2
UR-INDIC	1.0	4.4	58.9	99.0	95.7	69.8	19.5	65.5	17.3
	1.5	2.4	47.5	97.6	79.8	80.0	37.4	59.6	22.3
UR + BT	0.5	0.2	30.4	97.8	80.6	71.8	22.3	55.6	15.0
	1.0	0.1	27.0	95.4	73.6	77.6	29.6	53.5	16.9
UR-INDIC + BT	1.0	0.4	34.9	99.8	80.6	78.3	31.4	61.1	22.0
DIFFUR	1.0	2.1	50.6	99.9	91.6	80.8	25.2	72.7	20.9
	1.5	1.1	40.6	99.9	75.8	89.1	39.7	65.8	25.2
DIFFUR-INDIC	1.5	2.0	53.1	99.9	94.2	80.7	24.6	75.4	21.8
	2.5	0.9	41.4	99.9	75.6	86.1	36.9	64.6	24.3
DIFFUR-MLT	2.5	1.8	49.5	99.9	91.9	87.9	39.1	80.0	33.8
	3.0	1.0	40.0	99.1	73.0	92.1	56.5	65.3	35.0

Table 20: Performance breakdown of Bengali formality transfer by individual metrics described in Section 5.

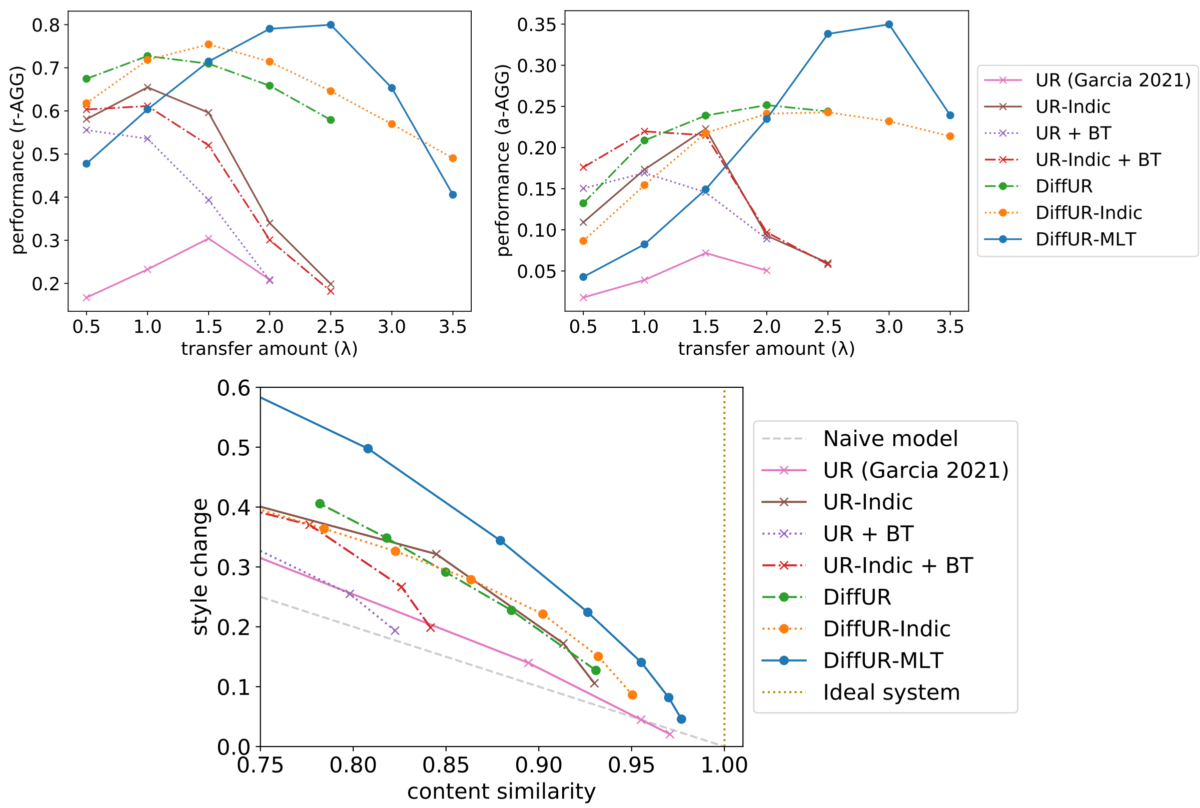

Figure 12: Variation in Bengali formality transfer test set performance & control for different models (see Table 20 for a individual metric breakdown of the models at the best performing λ). The plots show overall style transfer performance, using the r-AGG (top-left) and a-AGG (top-right) metrics from Section 5.5. We see the DIFFUR models outperform other systems across the λ range, and get best performance with the DIFFUR-MLT variant. We also see that DIFFUR models, especially with DIFFUR-MLT, lead to better style transfer control (bottom plot, closer to $x = 1$ is better), giving large style variation with λ without loss in semantics (X-axis).

Model	λ	COPY(↓)	1-g(↓)	LANG	SIM	r-ACC	a-ACC	r-AGG	a-AGG
UR (Garcia et al., 2021)	1.5	52.0	86.8	99.9	95.0	29.9	11.2	25.5	8.0
UR-INDIC	1.0	8.6	62.9	98.3	94.5	67.0	20.8	61.3	17.8
UR + BT	0.5	0.3	26.0	77.8	75.5	67.2	23.3	39.8	11.9
UR-INDIC + BT	0.5	1.6	40.6	99.9	82.3	73.9	26.8	59.2	19.1
	1.0	1.4	37.7	99.8	76.8	78.3	32.8	58.1	21.0
DIFFUR	1.0	3.0	47.4	99.8	87.9	80.3	30.5	69.2	23.6
	2.0	2.2	39.6	99.9	73.0	87.8	48.3	62.1	29.1
DIFFUR-INDIC	1.5	2.9	50.3	99.9	91.5	81.2	32.2	73.1	26.4
	2.0	2.3	45.2	99.9	82.7	85.1	42.3	68.5	29.3
DIFFUR-MLT	2.0	5.4	59.6	100	97.5	82.9	28.9	80.4	27.5
	3.0	2.1	42.7	99.1	71.7	92.6	63.4	64.5	39.4

Table 21: Performance breakdown of Kannada formality transfer by individual metrics described in Section 5.

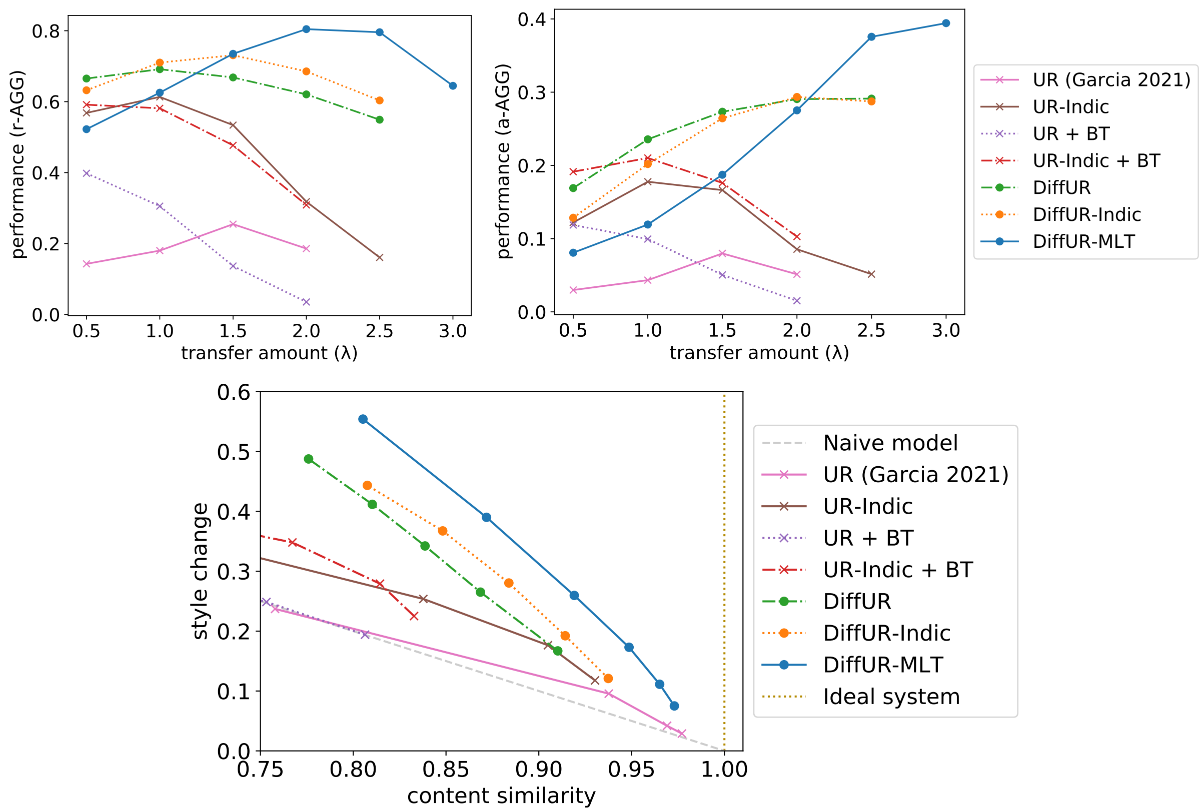

Figure 13: Variation in Kannada formality transfer test set performance & control for different models (see Table 21 for a individual metric breakdown of the models at the best performing λ). The plots show overall style transfer performance, using the r-AGG (top-left) and a-AGG (top-right) metrics from Section 5.5. We see the DIFFUR models outperform other systems across the λ range, and get best performance with the DIFFUR-MLT variant. We also see that DIFFUR models, especially with DIFFUR-MLT, lead to better style transfer control (bottom plot, closer to $x = 1$ is better), giving large style variation with λ without loss in semantics (X-axis).

Model	λ	COPY(↓)	1-g(↓)	LANG	SIM	r-ACC	a-ACC	r-AGG	a-AGG
UR (2021)	1.5	51.3	87.0	100	96.3	26.3	10.1	22.8	7.5
	2.0	35.0	68.2	99.9	73.0	45.4	28.6	20.7	8.4
UR-INDIC	1.0	10.4	64.5	98.8	94.3	65.6	20.2	59.8	16.7
	1.5	5.9	53.5	97.3	80.0	74.9	33.1	55.9	19.9
UR + BT	0.5	0.2	26.3	82.4	73.4	65.6	23.4	38.4	11.3
	1.0	0.1	19.8	74.9	64.7	71.2	31.6	33.1	11.6
UR-INDIC + BT	0.5	0.6	39.2	99.9	79.6	73.5	26.2	56.8	17.9
	1.0	0.5	36.1	99.7	74.0	78.5	35.9	56.0	22.2
DIFFUR	1.0	1.7	46.0	99.9	87.9	80.5	27.6	69.4	21.5
	2.5	0.9	36.0	99.8	68.4	90.2	47.2	59.9	27.1
DIFFUR-INDIC	1.0	2.4	50.1	99.9	91.7	78.7	28.7	71.0	23.7
	1.5	1.4	44.6	99.9	83.6	83.6	38.4	68.2	27.1
DIFFUR-MLT	2.0	3.8	55.8	99.9	95.7	84.0	31.2	79.8	28.6
	2.5	1.8	47.0	99.5	85.8	90.1	48.4	76.0	37.9

Table 22: Performance breakdown of Telugu formality transfer by individual metrics described in Section 5.

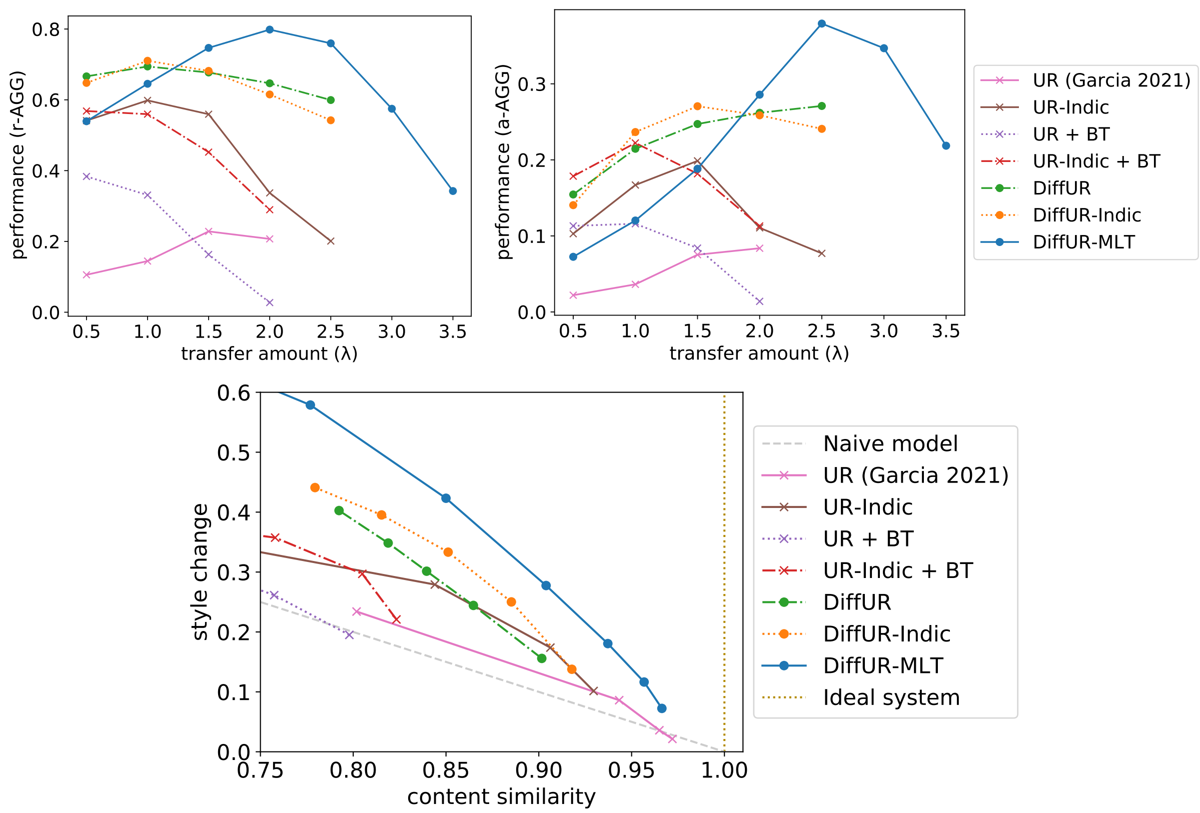

Figure 14: Variation in Telugu formality transfer test set performance & control for different models (see Table 22 for a individual metric breakdown of the models at the best performing λ). The plots show overall style transfer performance, using the r-AGG (top-left) and a-AGG (top-right) metrics from Section 5.5. We see the DIFFUR models outperform other systems across the λ range, and get best performance with the DIFFUR-MLT variant. We also see that DIFFUR models, especially with DIFFUR-MLT, lead to better style transfer control (bottom plot, closer to $x = 1$ is better), giving large style variation with λ without loss in semantics (X-axis).

Model	λ	COPY(↓)	1-g(↓)	LANG	SIM	r-ACC	a-ACC	r-AGG	a-AGG
UR (2021)	1.5	62.6	89.1	99.9	93.1	30.2	9.3	23.7	5.0
UR-INDIC	1.0	17.5	73.6	98.4	96.8	57.6	11.7	54.0	9.9
	1.5	10.9	62.7	96.9	85.4	67.0	19.2	53.0	10.7
UR + BT	0.5	0.5	34.3	87.3	77.6	69.1	17.8	46.3	9.8
	1.0	0.3	26.5	78.8	67.6	74.8	27.2	39.1	10.4
UR-INDIC + BT	0.5	1.9	47.4	99.9	87.1	68.1	22.0	57.7	16.8
DIFFUR	0.5	0.0	5.7	1.2	81.3	73.2	25.7	0.4	0.2
DIFFUR-INDIC	0.5	1.1	34.7	54.9	95.6	68.6	18.6	37.4	9.0
	1.5	0.4	24.2	46.0	74.7	78.5	40.0	29.2	13.0
DIFFUR-MLT	2.0	7.7	65.4	98.6	96.2	79.3	25.0	75.0	22.3
	2.5	4.5	54.6	95.1	85.5	86.0	45.8	69.8	33.1

Table 23: Performance breakdown of Gujarati formality transfer by individual metrics described in Section 5.

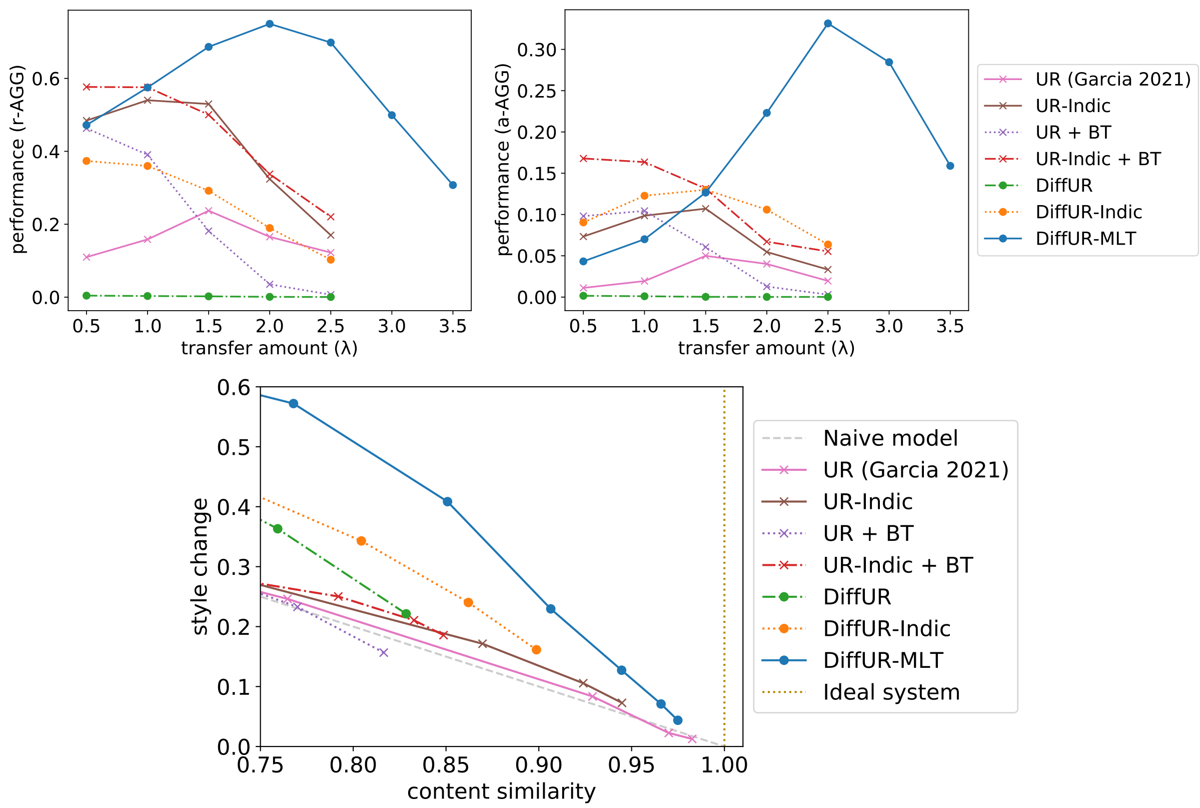

Figure 15: Variation in Gujarati formality transfer test set performance & control for different models (see Table 23 for a individual metric breakdown of the models at the best performing λ). The plots show overall style transfer performance, using the r-AGG (top-left) and a-AGG (top-right) metrics from Section 5.5. Note that Gujarati is a **zero-shot** language for DIFFUR models — no Gujarati paraphrase data was seen during training. We see that while the vanilla DIFFUR model performs poorly, the DIFFUR-INDIC is competitive with baselines and the DIFFUR-MLT variant significantly outperforms other systems. We also see that the DIFFUR-MLT variant lead to better style transfer control (bottom plot, closer to $x = 1$ is better), giving style variation with λ without loss in semantics (X-axis).

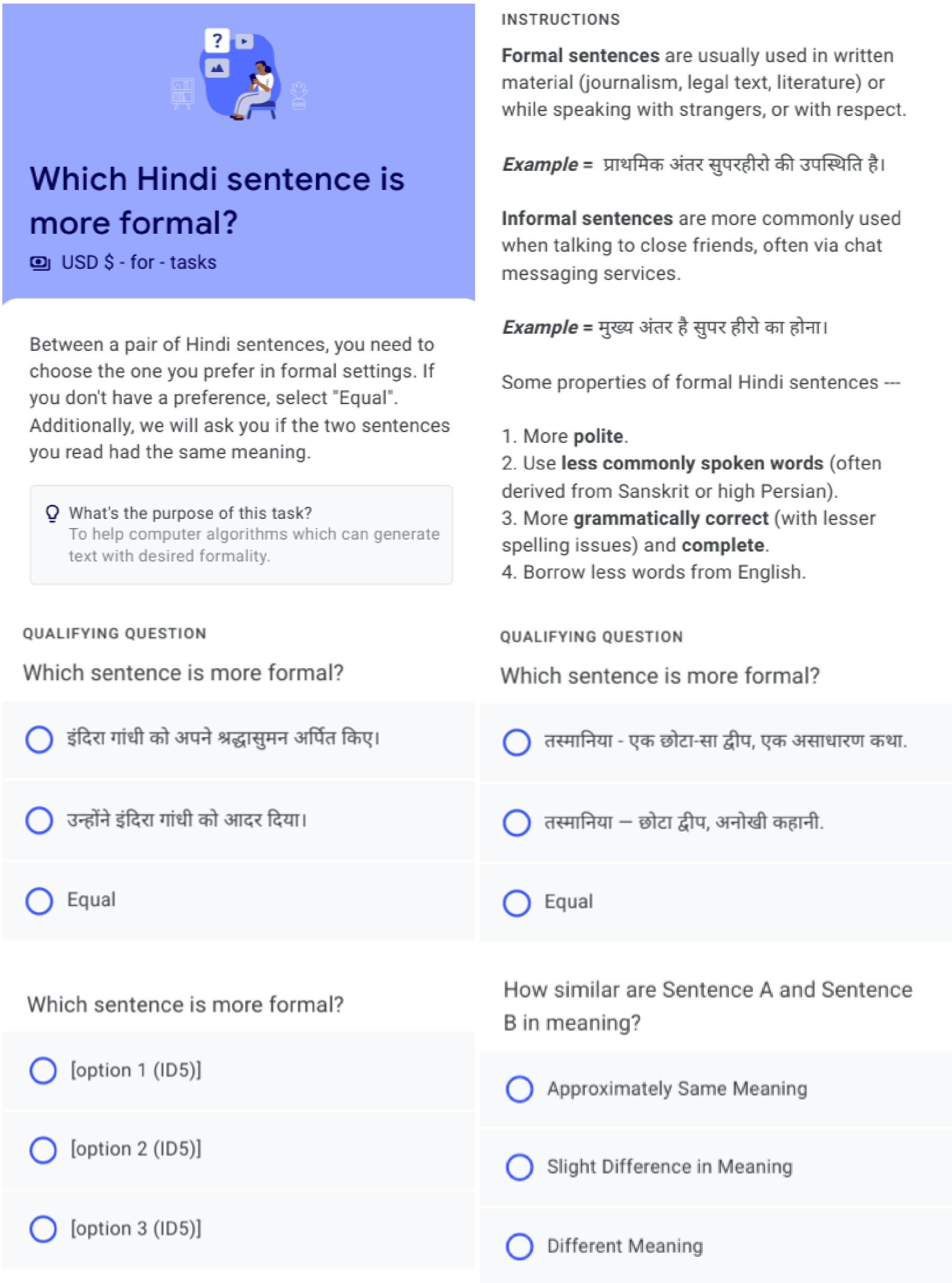

Figure 16: Our crowdsourcing interface on Task Mate, used to build our formality evaluation datasets (Section 5.1) and conduct human evaluations (Section 5.7). The first row shows our landing page and instruction set derived from our conversations with professional linguists. The second row shows our qualification questions for formality classification, and the third row shows templates for the two questions asked to crowdworkers per pair.