Style Constrained Generation with Few-Shot Exemplars

Vineeth Dorna
vdorna@umass.edu

Anmol Mekala
amekala@umass.edu

Siva Datta B
sbudaraju@umass.edu

Sneha Sree V
svavilapalli@umass.edu

Yashwanth Babu V
yashwanthbab@umass.edu

Abstract

When used for text generation, language models are biased towards repeating the tone and style of their training, and especially instruction finetuning steps. Style constrained generation counters this by allowing users to generate in a custom controllable style per their downstream tasks. This usually requires large amounts of style-labelled training data, either to finetune the generator, or in the training of a style classifier that guides the LM's generations during decoding. We show that leveraging T5's transfer learning in a SetFit contrastive learning framework helps provide effective guidance with just a handful style exemplars. This makes our style generation pipeline label-efficient by 4 orders of magnitude over prior works like FUDGE. Our implementation can be found at https://github.com/molereddy/SetFit-Style-Guidance/tree/main.

1 Introduction

The advent of the LLM era has seen the widespread adoption of language model generated text in online content generation, as in (Huyghebaert, 2023). However LLM-generated text, has an uncanny quality to it which outs it to human readers as machine generated (Johnson, 2023). In addition to this, there is the problem of specific end-users requiring a particular style for the text. Content publishers would like to maintain their previous tone, diction, and brand voice even after transitioning to using LLMs. In a related motivating example, one of the contributors to this project had built a custom OpenAI GPT to fill internship applications in the user's style, after providing around 10 diverse exemplars of style and content in the system context. Despite the power of GPT-4, and extensive prompting in an attempt to force the chatbot to adopt the style, style transfer was ineffective and it still tended to have its usual verbose and robotic tone.

These examples show that having a lightweight pipeline for generating texts with a particular style would have many useful downstream applications. In a realistic setup, this pipeline wouldn't involve usage of thousands of texts in the target style, and would just take a handful of target style exemplars as input. In addition we should avoid the cost and complexity of modifying our generator model itself.

1.1 Problem statement

Following (Yang and Klein, 2021)'s notation, style constrained generation involves modelling $P(X|I, a)$ where X represents the generated text, I represents any input context, including information on the task to be performed, and a is the desired style attribute.

In our setup, we choose $P(X|I)$ as a sentence paraphrase generation task, and the attribute a as *formality* of the sentence which we control during our generation process. We assume access to a small amount (5-40) of exemplars from a class to characterize the attribute a.

2 Our contributions

Our main task of implementing a classifier guided style generator which worked with only few-shot exemplars is completed as planned.

- Leveraged the SetFit contrastive learning framework (Jo et al., 2022) and transfer learning from a T5 sentence transformer (Ni et al., 2021a) to train our classifier with just a few exemplars.
- We implemented the FUDGE (Yang and Klein, 2021) classifier guidance for generation which uses a prediction rescoring approach, and *did not* use the PPLM (Dathathri et al.,

2019) framework of gradient based modification. FUDGE is simpler and modular, as we detail later.

- Evaluated – $P(a|X)$: quality of our style transfer with a classifier trained on large dataset, $P(X|I)$: content preservation using an NLI model. We *did not* get to evaluating the fluency $P(X)$ via perplexity, but we think $P(X|I)$ captures $P(X)$ to some degree.
- The nuanced nature of style limited our automated style evaluation results, so we also performed human annotations of random splits of our generated data.

3 Related work

The earliest works style constrained generation works (Ficler and Goldberg, 2017; Yu et al., 2017) focused on finetuning the generators to a target style. This is less effective because of the requirement of a large amount of target style exemplars and the finetuning dataset also might negatively affect the model's general capabilities. This line of work requires parallel text datasets and in absence of such data, (Krishna et al., 2020) use unsupervised style transfer by generating datasets via paraphrasers. We don't compare against this paper as we avoid finetuning, but this paper *shares the core paraphrasing task* with our setup.

GENRE (De Cao et al., 2021), restricts generation to certain entities via a trie structure that is used to contained the entities, narrowing down the next tokens. However, this simplistic approach faces limitations when tackling more intricate and subjective constraints, such as sentiment and styles like formality.

Approaching controlled generation as a multi-objective optimization problem, MUCOCO (Kumar et al., 2021) focuses on balancing various objectives such as semantic similarity with the input sentence, achieving the target style, and ensuring the fluency of the generated text. This approach optimizes across multiple dimensions of the generation simultaneously, using lagrange multipliers and gradient descent iteratively. While we found this work very interesting and performant, we resorted to using simpler frameworks outlined below.

Few-shot and zero-shot learning approaches have become very effective with recent large language models, which encode large amounts of training data across styles in their parameters. These patterns can be unlocked during generation via different prompts in the context. (Reif et al., 2021; Zhou et al., 2023) simply preprend a prompt in zero-shot and few-shot settings and achieve effective style generation. But our setup *shares the "few-shot" aspect* of these works. As (Jo et al., 2022) discuss, these hand crafted prompts are brittle and require billion-parameter LMs to achieve high accuracy. We preferred to focus on controlling the decoding step ourselves instead of the black-box prompt engineering of these papers.

Another class of approaches modify model activations via gradients before generation. Two of these are DEXPERTS (Liu et al., 2021) and PPLM (Dathathri et al., 2019). DEXPERTS trains "expert" and "anti-expert" generative LMs and adjusting the next token x_{t+1}'s distribution accordingly. PPLM also comes under the theme of classifier guided generation methods, which we discuss next.

PPLM and FUDGE (Yang and Klein, 2021) use *classifier guided* generation approaches. PPLM incorporates a classifier head trained to predict attribute a for a given sequence $x_1, x_2 \cdots x_t$. During generation, it leverages the generated sequence $x_1, x_2 \cdots x_t, x_{t+1}$ and the trained classifier head and sends gradients of the predicted a to update the hidden representation H_t of the generator and then regenerate $x_t + 1$. This method is complicated by the involvement of hidden state changes and gradient flows. Another limitation is that the generation is greedy, focusing on a local "next-token style" instead of a global "full-sentence" style.

We've borrowed aspect of our problem setup's pipeline from different parts of the preceding paragraphs, chief among which is the FUDGE framework we outline next.

3.1 How FUDGE works

FUDGE initially generates a distribution for the next token x_{t+1}, then forms multiple candidates based on the top-k tokens and utilizes a classifier to estimate the probability for constraint a given the input sequence $x_1, x_2 \cdots x_t, x_{t+1}$. One just needs to have access to a generator's predictions during decoding and then any attribute classifier model can be plugged onto it to guide the generation. FUDGE also belongs to a family of weighted decoding methods (Ghazvininejad et al., 2017; Holtzman et al., 2018; Shen et al., 2019)

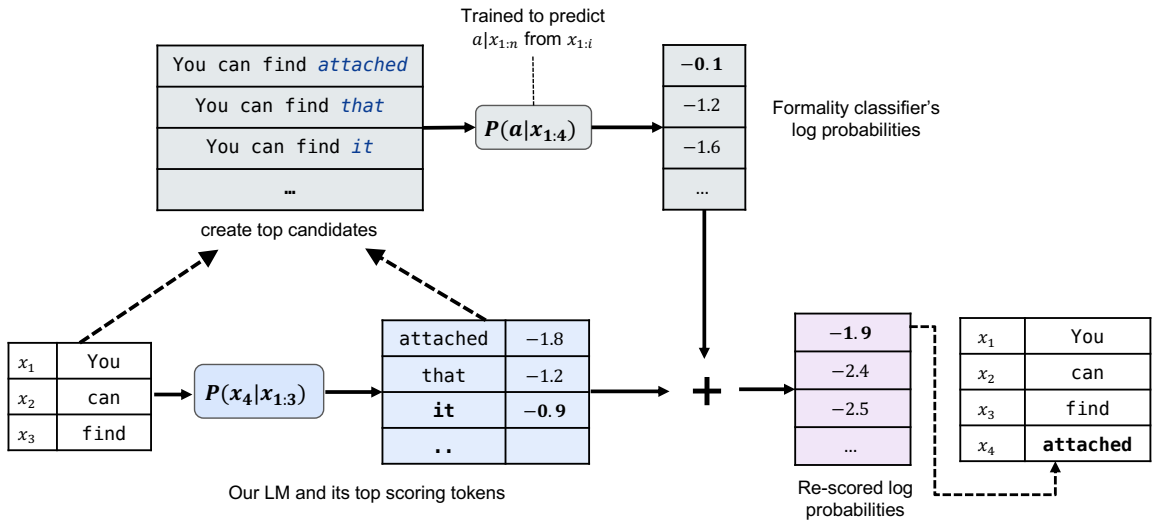

Figure 1: FUDGE generation: Here, even though the three candidates look roughly equally formal, what matters is the *future* generation's formality. "*You can find attached...*" is a prefix usually preceding a workplace email, which is likely to be more formal than the others. FUDGE promotes *attached* based on this potential formality.

that assume access only to the decoding stage logits and rework the weights using different approaches.

Final scores $P(X|a, I)$ are modelled by the equation

$$P(X|a, I) = P(X|I) \cdot P(a|X)^{\lambda} \quad (1)$$

where $P(a|X)$ is the classifier guidance, and $P(X|I)$ is the original score and λ is the guidance strength.

Compared to similar methods, notable in FUDGE is the usage of a classifier to predict the style of a potential *future* completed sentence of the prefix it sees now, than greedily predicting current prefix. That is, FUDGE's classifier $P(a|x_{1:i})$ aims to model $a_{true}|x_{1:n}$ instead of $a_{true}|x_{1:i}$. This is done by training the classifier with label a for all the prefixes $\{x_{i:j}, j \in 1 \cdots n\}$ of a style exemplar text, whether or not a prefix $x_{i:j}$ possesses the attribute a by itself.

We found FUDGE's framework and method to be simple and modular, and we adopted it for our specific task of paraphrase generation. The method is illustrated in figure 1.

4 Our modification to FUDGE

Our approach seeks to diverge from using extensively finetuned classifiers for guidance, by restricting our access to a set of minimal (few-shot) attribute labels. Our guide model is robust and effectively steers the generation process, even though the data it has seen is quite limited.

We see two pathways of learning accurate and robust style representations in a label efficient manner. One is *transfer-learning*, where the existing rich representations learnt by a pretrained LM can be easily modified while adapting to downstream tasks. In the classification setup, another powerful way of improving label efficiency is to contrast pairs of examples from the training data. Creating pairs helps leverage more labels as the number of possible examples in our loss function grows quadratically. *Contrastive learning* shifts representations of exemplars from within a class together and exemplars from different classes away from each other to force representation clustering.

4.1 SetFit

The SetFit few-shot learning framework (Tunstall et al., 2022), combines these two pathways into a label-efficient and prompt-free finetuning framework by leveraging contrastive training and Sentence Transformers for powerful embedding models to transfer learn from. This is particularly effective in the few-shot setup, where, across a wide variety of tasks, a handful of exemplars are enough for high accuracy classification. This makes SetFit several orders of magnitude label-efficient to train and this optimization on the FUDGE pipeline (detailed in § 3.1) is our main algorithmic contribution.

We start with pretrained Sentence Transformer models that already have rich embeddings that can

Dataset	Sample sentences
PAWS	The dude moved to Connersville, settled in Indiana, and kept on lawyering. The second series was more favorably regarded by critics in comparison to the first.
GYAFC	Formal: Not all music in the genre of rap is terrible. Informal: yea, um, this is a Korean singer, who is really good.
Daily Dialog	His nose is out of joint because we forgot to invite him to the party. So, I finally got myself new skates.
SQUINKY	Score 2.8: I would trust the social workers to make the appropriate case by case determination. Score -3: west indies for sure would win!!!

Table 1: Examples from the different datasets

be efficiently modified for our task with little feedback from labels. SetFit first performs contrastive finetuning of these *embedders* with a triplet loss, over positive and negative pairs created from the few-shot exemplars. These embeddings now cluster according to our target attribute.

The triplet loss formulation for the loss $L(X, P, N)$, for an embedding f and margin α, given a positive X, P pair and the negative X, N pair is given as:

$$\max(\|f(X) - f(P)\|_2 - \|f(X) - f(N)\|_2 + \alpha, 0) \quad (2)$$

where A is an anchor input, P is a positive input of the same class as A, N is a negative input of a different class from A, α is a margin between positive and negative pairs, and f is an embedding.

In the next step a single classification head is finetuned for our downstream attribute classification task, transforming our LM into a effective classifier. Having separable clusters helps learn the decision boundary easily.

5 Experiments

5.1 Datasets

We evaluate our style-paraphrasing method on sentences from 3 datasets. Table 1 presents few examples for each dataset. We sample 500 examples from each dataset to average our results over.

1. **Daily Dialog** (Li et al., 2017): contains 11k examples of human written dialogues in a multi-turn setting which reflects patterns from daily communications.
2. **GYAFC** (Rao and Tetreault, 2018): a 100k sized dataset of labelled formal and informal sentences, created for use in style transfer.
3. **PAWS** (Zhang et al., 2019): a labeled paraphrasing dataset of 53k example sentence pairs made from Wikipedia and Quora.

Pre-processing We did not have to pre-process these datasets except to create prefixes from each sentence in our classifier training data, as discussed in § 3.1). This prepares our classifiers to make *future* style predictions and also improves classifier quality as regulariser via data augmentation.

5.2 Metrics

We evaluate the generations based on metrics which capture two main themes.

Style evaluation: We quantify the intensity of style in a text using available trained neural models.

- **FC – Formality Classification:** We utilize a DeBERTa-based classification model (Dementieva et al., 2023), fine-tuned on the GYAFC dataset, to quantify the formality. We report FC as the average probability of a text being classified as formal. This models $P(a|X)$, which is our style constraint.
- **FS – Formality Score:** We also report a continuous score based on training on the SQUINKY formality dataset (Lahiri, 2015), which has a float rating scale from -3 (informal) to 3 (formal). To calculate FS, we trained a BERT-based regression model on SQUINKY and the average output of the BERT regression model on the

texts as FS. This is also a measure that correlates with $P(a|X)$.

CP – Content Preservation: this measures the paraphrasing quality via the semantic similarity between two pairs of texts to ensure that our style generation does not damage the original sentence. We employ a RoBERTa-based NLI model from (Reimers and Gurevych, 2019) trained to report the average of Similarity(X, I), which correlates with our intention of capturing $P(X|I)$.

Human evaluation: We also perform and report human evaluations for both styles FS_{human} and content preservation CP_{human} on 20 examples. For FS_{human}, we ask the annotator to label the style form -3 (informal) to 3 (formal), to match SQUNKY's formality scale. For CP_{human} we ask the annotator to give rating from 1 to 5 based on the content preservation and we report a 0-100 scaled version.

5.3 Baseline

We compare our method with FUDGE (Yang and Klein, 2021), where we train a 'gold' classifier on the entire GYAFC training corpus (of 50k examples per class). This is in contrast to our SetFit model's training set, which utilizes just a few examples. This FUDGE baseline could be also termed a "skyline", because, being trained on more labels, the FUDGE classifier is likely to inherently be better than our fewshot SetFit classifier. Our aim is to achieve similar performance and acceptable style quality, with a lesser number of examples.

5.4 Implementation details

Task: We perform our style transfer algorithm on paraphrasing tasks due to the availability of good quality models and data for this task.

Paraphraser: Existing paraphrasing models trained on non-synthetic paraphrasing datasets (e.g., PAWS (Zhang et al., 2019)) show poor performance due to the low quality of the data, where the input and the paraphrase target have very little difference. Tihs resulted in the paraphraser barely paraphrasing most sentences. In our approach, we employ a T5-based seq2seq paraphrasing model trained on a synthetic dataset (Vorobev and Kuznetsov, 2023) as our base generative model. This model has been trained on a diverse range of texts paraphrased by ChatGPT, sourced from multiple platforms including Quora, SQuAD, CNN etc., and performed at a much higher quality than our PAWS-trained models.

FUDGE setup: We re-implemented the FUDGE training and classification setup by ourselves as the existing implementation was quite outdated. Our implementation of FUDGE differs somewhat from the original version in these respects: original version used a 10M sample synthetic dataset to train LSTMs from scratch. We utilized a T5 based classification model and finetuned it on the GYAFC training set.

SetFit training: We trained a T5 based sentence transformer (Ni et al., 2021b) using the SetFit framework of contrastive learning on 5-40 random exemplars from GYAFC of each of the two styles. The SetFit classifier was trained with a batch size of 20, with a cosine similarity loss. Since the existing SetFit framework models don't take token ids as inputs, we forked and extended the implementation to allow for this functionality in our version of SetFit.

Tokenizer challenges and T5: A major challenge in our implementation was the fact that for our classifier to re-score the generator's token sequences, the tokenizers of both models had to be exactly matched. This forced us to use versions of T5 for all our guidance classifiers (both in the FUDGE baseline and our method) and paraphrase generators. T5 was the only model which had a generator version and also a SetFit Sentence-Transformer framework available. To have a perfect plug and play setup of FUDGE across any classifier and generator model, we'd have to implement a cross-tokenizer aligner module, but our project scope restricted us from picking this up.

Guidance strength: While re-scoring our generators logits with the classifier scores, it is important to decide the strength of the classifier guidance λ (from eq. 1). We chose λ as 20 for both formal and informal scoring as the standard for our experiments. We perform ablations for this value in table 6.

Other technical details We utilize HuggingFace Transformers (Wolf et al., 2019) for most of our implementations. All our experiments are performed in NVIDIA QUADRO RTX 8000. For training our T5 classifier on the entire training set to reproduc FUDGE we needed 7h, while our

Dataset	Method	No Guidance			Informal Guidance			Formal Guidance		
		FC	FS	CP	FC(↓)	FS(↓)	CP(↑)	FC(↑)	FS(↑)	CP(↑)
Daily-Dialog	FUDGE	89.2	-0.28	80.8	**14.05**	-1.36	**43.0**	**98.46**	**0.24**	65.8
	Ours				24.55	**-1.63**	39.2	92.45	-0.19	**73.8**
GYFAC	FUDGE	93.18	-0.16	78.4	**12.79**	-1.20	35.19	**98.74**	**0.18**	60.8
	Ours				22.37	**-1.45**	**39.2**	95.34	-0.08	**70.8**
PAWS	FUDGE	99.53	0.92	80.4	33.13	0.28	**20.8**	98.99	**0.91**	54.8
	Ours				**26.83**	**-0.05**	12.2	**99.36**	**0.91**	**64.4**

Table 2: Evaluation of our generated text. FC denotes a *formality score*, based on a GYAFC-trained model's classifications. FS also measures formality, but uses a regression model trained on the Pavlick formailty dataset, which scores formality in the range $(-3, 3)$. CP is the NLI-based semantic similarity of the paraphrased sentence compared to the original.

Dataset	Example	FUDGE	Ours
PAWS	Holly was musically influenced by Elton John.	Although Holly identified with Elton John, she also incorporated his musical influences into her work.	Elton John had a significant impact on Holly's musical tastes, as Holly was influenced by the latter.
GYAFC	I don't think that page gave me viruses.	Although, I believe I was safe from viruses on that page.	I've never encountered a virus on that page that I suspect gave me the wrong type of information.
Daily dialog	What position do you play?	Could you describe your playing position?	What is your preferred position?

Table 3: Examples and comparision against baseline in formal generation

SetFit based classifier was trained within 1h. The T5 classifier used to reproduce FUDGE and the style-scoring BERT regressor were trained with a batch size of 48 and weight decay of 1e-2, 500 warmup steps and learning rate scheduling left to the HuggingFace trainer. Best model was chosen based on validation loss after 3 epochs.

6 Results

Table 2 presents the main results where we evaluated the generations of our version of FUDGE (Ours) against the original label-inefficent baseline/skyline (FUDGE) across datasets and styles (See § 5.3 for why we also call FUDGE a skyline). Table 3 presents the comparison of *formal* and Table 4 informal generated text by both methods. We observe that:

Style guidance modifies style as directed: Compared to *No Guidance*, adding a style guidance increases the FC/FS scores for *Formal Guidance* and decreases the same scores for *Informal Guidance*. This shows that style transfer via weighted decoding works. We also observe that across datasets, even without style transfer, sentences tend to be classified as formal, because informal exemplars seem to have unusual words and linguistic mistakes, which a base paraphraser would not produce. We think that the "clean" sentences produced by the base paraphraser are then simply classified as formal by our classifiers.

Main thesis — Our method is comparable to FUDGE: As anticipated in § 5.3, our method is marginally behind FUDGE , with no significant trend across datasets. Despite using just a tiny amount of training samples, this comparable performance is expected: as long as our classifier is good enough at future style discrimination, it should be effective at style transfer. The efficacy of our classifier is evident from the SetFit training, which concluded with low style discrimination loss on the test data. Consequently, the quality and style of our generated outputs are expected

Dataset	Example	FUDGE	Ours
PAWS	Holly was musically influenced by Elton John.	Elton John wsporite music by influenced Holly a bit whither thailand.	Eagan Holly drew musical symscores due Elton js and a fusion musica of Endri & Holly's lyrical strands on the blues album.
GYAFC	I don't think that page gave me viruses.	virus on that page, I don't think I got any on.	virus because I don't think that page gave me that kind of information.
Daily dialog	What position do you play?	I'm not even a player but where do you usually play ido and if you were at dvr. ask yourself :)?	Whos in what position do you play?

Table 4: Examples and comparision against baseline in informal generation

Method	No Guidance		Informal Guidance		Formal Guidance	
	FS_{human}	$CP_{human}(\uparrow)$	$FS_{human}(\downarrow)$	$CP_{human}(\uparrow)$	$FS_{human}(\uparrow)$	$CP_{human}(\uparrow)$
FUDGE	1.3	71.25	**-2.15**	**51.25**	**1.85**	66.25
Ours			-2	22.5	**1.85**	**70**

Table 5: Human evaluation results on the Daily Dialog dataset. The metrics are analogous to those from other tables and are described in section 6.

to directly depend on the quality of the classifier. We do observe that for *formal*, we tend to be worse than FUDGE and the opposite for *informal* guidance. We hypothesize that this be attributed to classifier calibration, which can, for example, make a classifier's *formal* signal stronger than another's despite having the same guidance strength hyperparameter λ. If one finds the extent of style transfer unsatisfactory, tuning λ (as described in § 6.1) or calibrating the classifier would be a workaround for improving style without damaging quality.

Style guidance damages content quality: Compared to *No Guidance*, *Formal* and *Informal* guidance in decoding damages the Content Preservation (CP) scores. This is expected since we damage the optimum paraphrase in during re-scoring via weighted decoding. We are modelling $P(X|a, I)$, instead of $P(X|I)$, so $P(X|I)$ of the original task will necessarily fall. But we observe that, by our metrics, the damage is quite significant, and especially so for *informal* guidance. This makes sense as informal styles are noisy and tend to cover up the semantic content. We hypothesize that the extent of damage is also explained by the quality of our guidance classifiers described previously. Since making mistakes in the sentence is correlated with informality, these guides might sometimes artificially introduce semantic mistakes during style transfer. This is an inherent limitation of choosing informality setup transfer in our setup. More analysis of these metrics via human and error analysis is detailed in the coming sections.

While identifying the bias in using the metric FC where both our guidance classifier and evaluator model trained on same datasets, we also calculate FS which has a richer gradation based scoring. However limitation of FS score is the the model is trained on dataset of size 10x smaller than GYAFC. We could see that our model, in terms of FS, consistently outperforms FUDGE when guided for informal style.

Table 9 shows where our method faces its limitations while generating texts for short sentences. In case of informal generation, it generates non relevant characters to the end and also alters the meaning of the text. Whereas for formal it tends to alter only the meaning of the sentences for short texts. It also is particularly worse for small sentences as expected.

Human Evaluation As we observed little correlation between FC and FS scores considering both the styles (potentially due to calibration is-

Guidance (λ)	Informal Guidance			Formal Guidance		
	FC(↓)	FS(↓)	CP(↑)	FC(↑)	FS(↑)	CP(↑)
0	89.2	-0.28	80.8	89.2	-0.28	80.8
10	26.96	-1.61	**42.4**	91.75	-0.22	**74.6**
15	24.74	**-1.63**	41.8	**92.47**	-0.20	73.2
20	**24.55**	**-1.63**	39.2	92.45	**-0.19**	73.8

Table 6: Ablation over style guidance λ parameter for our SetFit based FUDGE method: on Daily Dialog.

# Shot	No Guidance			Informal Guidance			Formal Guidance		
	FC	FS	CP	FC(↓)	FS(↓)	CP(↑)	FC(↑)	FS(↑)	CP(↑)
5	89.2	-0.28	80.8	61.6	-1.13	26.0	92.8	**-0.13**	68.4
10				57.45	-1.10	22.6	**92.81**	**-0.13**	68.2
20				28.75	-1.60	**44.0**	91.71	-0.18	**75.8**
40				**24.55**	**-1.63**	39.2	92.45	-0.19	73.8

Table 7: Ablation over number of style labels for SetFit: Daily Dialog

sues with the evaluator models), we resorted to human evaluation as described in § 5.2. We hoped this would capture formality more intuitively. We perform human evaluation on 20 examples generated from Daily Dialog and present them in Table 5. We can see that our method is able to perform comparably well in terms of FS_{human} and show improvements in CP_{human} during formal guidance. Using human eval gives more intuitive results when compared to the FC and FS metrics, since subjectivity of style was major factor here. The CP metrics align with the same trends as seen in automated evaluation.

6.1 Ablations

Guidance strength (λ): To study the effect of guidance signal on generations, we experiment with various λ values in Table 6. As we increase the λ we observe that the generation style is drifted towards the respective guided style with minimal decrease in CP.

 # Few-shot examples In Table 7 we ablate over performance on varying number of few-shot exemplars in training our SetFit classifier model. We could see that as we increase the fewshot examples the FS, FR and CP scores improve, particularly more evident in informal guidance. But we clearly observe that adding more examples isn't that necessary and soon only provides diminishing returns. This further substantiates our belief in SetFit's fewshot discriminative ability.

Effect of future discrimination FUDGE's main theme was about future discrimination. In their study, they did not ablate this on aspect solely divorced from their other interventions. We perform ablation on this aspect by comparing against weighted decoding of a classifier trained without usage of prefixes in training. In Table 8, we observe a clear trend of improvement with *future* discrimination via prefix training.

7 Individual contributions

- **Vineeth Dorna**: SetFit idea, project design, implementation of classifier guidance, experimentation, writing.
- **Anmol Mekala**: Project design, writing, implementation of baseline, evaluators & token-level SetFit.
- **Yashwanth B. Vunnam**: SetFit training under various setups and ablations, evaluators & final experimentation.
- **Siva Datta B**: Dataset setup, paraphraser experimentation, GPT-4 eval, human eval.
- **Sneha Vavilapalli**: Dataset setup, SetFit experimentation, human evaluation and report writing.

8 Conclusion

In our study, we explored a weighted decoding approach in guiding the generation process toward specific styles, and optimized the label efficiency

Classifier focus	No Guidance			Informal Guidance			Formal Guidance		
	FC	FS	CP	FC(↓)	FS(↓)	CP(↑)	FC(↑)	FS(↑)	CP(↑)
Greedy	89.2	-0.28	80.8	40.19	-1.54	**40.6**	93.02	-0.21	**74.8**
Future				**24.55**	**-1.63**	39.2	92.45	**-0.19**	73.8

Table 8: Ablation over the usage of prefix labels in classifier training: Daily Dialog dataset. FUDGE (*Future*) uses prefixes of the style exemplars in training to find a *future* style estimation. While prior approaches, exemplified by *Greedy*, focus only on the next token's optimum style label.

Original sentences	Generated sentence
Say, Jim, how about going for a few beers after dinner? ⇒ Informal	How you guys is doing? I need to go out for a couple beers afternu doz. Jim jibe qld saj na me li koo w c\textasciicircum c\textasciicircum hu x dz?
Not yet. ⇒ Informal	Preg*f*É...
Do you like cooking? ⇒ Informal	Cooky?
Do you like traveling, Kathy? ⇒ Informal	Kartya... do you like to go on errands?
I'm afraid not. ⇒ Formal	I'll be shocked if I don't.
See you ⇒ Formal	Good luck.
sounds good! Let's go! ⇒ Formal	Those are some really good sounds!!

Table 9: Example sentence for limitations of our method

of the classifier used in this process. Our results demonstrate that we can achieve this goal with minimal examples by exploiting SetFit's ability to capture stylistic nuances with only a handful of examples. Despite achieving success in style transfer, the simplicity and modularity of this rescoring setup we chose inherently limited our task quality for the sake of style. Our ablations showed that this generation is easily steerable, label-efficient and benefits from "looking-ahead" during discrimination. During evaluation we observed that the subjective nature of style causes disagreement and some less intuitive results for automatic evaluators. We then turned to human evaluation which provided intuitive results while continuing to agree with the broad themes of our automated evaluations.

In terms of the project, we learnt a lot about tokenization and how difficult it makes life for NLP practitioners. We also gained good experience training and adapting a wide variety of models across classification, regression tasks, seq2seq and encoder-only architectures. We also experimented on a wide variety of datasets and learnt about issues with dataset quality and scale.

9 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - No.

References

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. (2019). Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

De Cao, N., Izacard, G., Riedel, S., and Petroni, F. (2021). Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dementieva, D., Babakov, N., and Panchenko, A. (2023). Detecting text formality: A study of text classification approaches. In Mitkov, R. and Angelova, G., editors, *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 274–284, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Ficler, J. and Goldberg, Y. (2017). Controlling linguistic style aspects in neural language generation. *arXiv preprint arXiv:1707.02633*.

Ghazvininejad, M., Shi, X., Priyadarshi, J., and Knight, K. (2017). Hafez: an interactive poetry generation system. In

Proceedings of ACL 2017, System Demonstrations, pages 43–48.

Holtzman, A., Buys, J., Forbes, M., Bosselut, A., Golub, D., and Choi, Y. (2018). Learning to write with cooperative discriminators. *arXiv preprint arXiv:1805.06087*.

Huyghebaert, C. (2023). Lessons learned building products powered by generative ai. Medium. Accessed: 2024-03-08.

Jo, U. E. S., Tunstall, L., Bates, L., Korat, D., Pereg, O., and Wasserblat, M. (2022). Setfit: Efficient few-shot learning without prompts. HuggingFace Blog.

Johnson, A. (2023). Chatgpt and the uncanny valley of language. Medium. Accessed: 2024-03-08.

Krishna, K., Wieting, J., and Iyyer, M. (2020). Reformulating unsupervised style transfer as paraphrase generation. *arXiv preprint arXiv:2010.05700*.

Kumar, S., Malmi, E., Severyn, A., and Tsvetkov, Y. (2021). Controlled text generation as continuous optimization with multiple constraints. *Advances in Neural Information Processing Systems*, 34:14542–14554.

Lahiri, S. (2015). SQUINKY! A corpus of sentence-level formality, informativeness, and implicature. *arXiv preprint arXiv:1506.02306*.

Li, Y., Su, H., Shen, X., Li, W., Cao, Z., and Niu, S. (2017). Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*.

Liu, A., Sap, M., Lu, X., Swayamdipta, S., Bhagavatula, C., Smith, N. A., and Choi, Y. (2021). Dexperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*.

Ni, J., Ábrego, G. H., Constant, N., Ma, J., Hall, K. B., Cer, D., and Yang, Y. (2021a). Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *CoRR*, abs/2108.08877.

Ni, J., Abrego, G. H., Constant, N., Ma, J., Hall, K. B., Cer, D., and Yang, Y. (2021b). Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*.

Rao, S. and Tetreault, J. (2018). Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In Walker, M., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Reif, E., Ippolito, D., Yuan, A., Coenen, A., Callison-Burch, C., and Wei, J. (2021). A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Shen, S., Fried, D., Andreas, J., and Klein, D. (2019). Pragmatically informative text generation. *arXiv preprint arXiv:1904.01301*.

Tunstall, L., Reimers, N., Jo, U. E. S., Bates, L., Korat, D., Wasserblat, M., and Pereg, O. (2022). Efficient few-shot learning without prompts. *arXiv preprint arXiv:2209.11055*.

Vorobev, M. K. V. and Kuznetsov, M. (2023). A paraphrasing model based on chatgpt paraphrases. *A paraphrasing model based on ChatGPT paraphrases*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yang, K. and Klein, D. (2021). Fudge: Controlled text generation with future discriminators. *arXiv preprint arXiv:2104.05218*.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Zhang, Y., Baldridge, J., and He, L. (2019). PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*.

Zhou, W., Jiang, Y. E., Wilcox, E., Cotterell, R., and Sachan, M. (2023). Controlled text generation with natural language instructions. In *International Conference on Machine Learning*, pages 42602–42613. PMLR.