Meta-Tuning LLMs to Elicit Lexical Knowledge of Language Style

Anonymous ACL submission

Abstract

Language style is often used by writers to convey their intentions, identities, and mastery of languages. In this paper, we show that current large language models struggle to capture some of the language styles without fine-tuning. To address this challenge, we investigate whether LLMs can be meta-trained based on representative lexicons to recognize new language styles that they have not been fine-tuned on. Experiments on 13 established style classification tasks, as well as 63 novel tasks generated using LLMs, demonstrate that meta-training with style lexicons consistently improves zero-shot transfer across styles. Code and data to reproduce our experiments will be released upon publication.

1 Introduction

The style of a text refers to unique ways authors select words and grammar to express their message (Hovy, 1987). It can provide insights into social interactions and implicit communication. A notable example underscoring the importance of studying linguistic style used in communication is the analysis of body camera footage and transcripts (Voigt et al., 2017), where police officers have been found to use less respectful language towards black people than white people. Moreover, the open-ended and ever-evolving nature of language styles (Xu, 2017; Kang and Hovy, 2021) motivates the need for zero-shot classification, as it is costly to annotate data for every possible style in every language.

This leads to a natural question: *can recently developed instruction-tuned language models do well in identifying the style of texts without labeled data?* As we will show in the paper (§3.2), this remains a challenge, even though these models have demonstrated impressive zero-shot performance on many other tasks (Chung et al., 2022; Ouyang et al., 2022). On the other hand, before the paradigm in NLP shifted to pre-trained language models, lexicons of words that are stylistically expressive were commonly used as important lexical knowledge (Verma and Srinivasan, 2019) in rule-based (Wilson et al., 2005; Taboada et al., 2011), feature-based (Mohammad et al., 2013; Eisenstein, 2017), and deep learning models (Teng et al., 2016; Maddela and Xu, 2018) for style identification. Many lexicons have been developed for varied styles, such as politeness (Danescu-Niculescu-Mizil et al., 2013), happiness (Dodds et al., 2015), emotions (Mohammad and Turney, 2010; Tausczik and Pennebaker, 2010), etc. This leads to another research question: *can we leverage lexicons during instruction fine-tuning of large language models (LLMs) to improve their understanding of language style?*

In this paper, we examine the effectiveness of fine-tuning LLMs to interpret lexicons that are provided as inputs to elicit latent knowledge (Kang et al., 2023) of language styles that were acquired during pre-training. We first compile a benchmark of 13 diverse writing styles with both annotated test sets and style-representative lexicons. Using this benchmark, we show that **meta-tuning with lexicons** enables different pre-trained LLMs to generalize better to new styles that have no labeled data. For example, meta-tuning LLaMA-2-7B (Touvron et al., 2023) on seven styles can improve the average F1 score on a separate set of six held-out styles by 12%, and by 8% over a general instruction-tuned model, LLaMA-2-Chat.

To further verify the capability of LLMs to generalize to novel styles using lexicons as the only source of supervision, we generated a diverse set of 63 unique writing styles with examples (§4) using an approach inspired by self-instruction (Wang et al., 2023). We demonstrate that using a small lexicon of as few as five words can effectively improve generalization to new styles. We found it helpful to replace class names with random identifiers when meta-training models with lexicons, which prevents models from ignoring the lexicons and

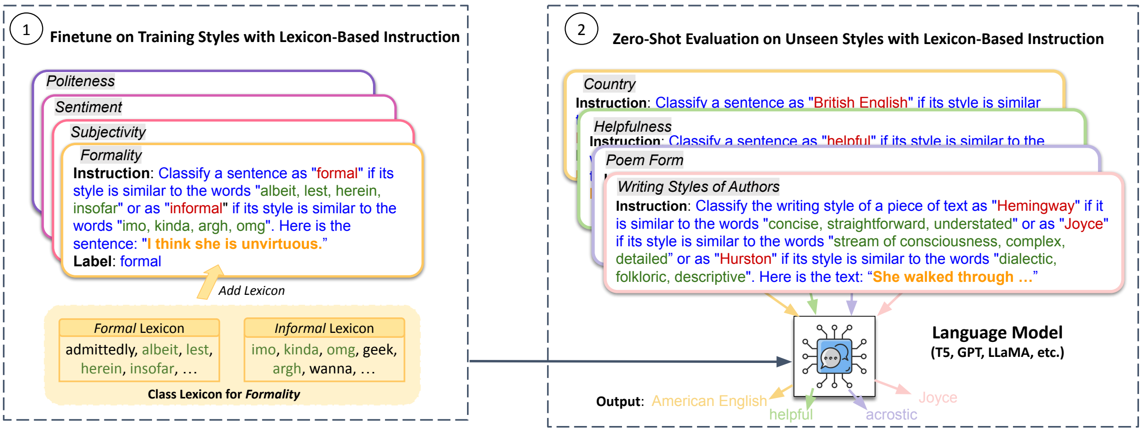


Figure 1: Overview of using lexicon-based instructions for cross-style zero-shot classification. It consists of two steps: (1) instruction tuning the model on training styles; (2) evaluating the learned model on unseen target styles zero-shot. A lexicon-based instruction is composed of instruction, class names, lexicons and an input.

simply memorizing source styles' class names. In addition, we show that when combined with **meta in-context learning** (Min et al., 2022a; Chung et al., 2022), incorporating lexicons can significantly reduce variance.

We will make our data, along with code to reproduce our experiments available for publication.

2 Meta-Tuning for Style Generalization

We investigate the capabilities of LLMs to interpret language styles using lexical knowledge, and identify text that is representative of the associated styles. We compare lexicon-based instructions with other methods in the zero-shot setting, and further explore a few-shot setting. To study the effectiveness of meta-tuning with lexicons, in generalizing to various writing styles, we first consider a set of thirteen styles, where high-quality annotated data is available. Later, in §4, we evaluate the ability of lexicon-instructed models to generalize to 63 novel LLM-generated styles.

2.1 Problem Definition and Approach

Given an input text and style with pre-defined classes $C = \{c_k\}_{k=1}^{|C|}$, we present the language model with lexicon-based instructions, by instantiating lexicons (i.e., a list of words or phrases that are representative of each class c_k) in a pre-defined instruction template (see templates in Table 20). A language model is expected to predict one of the classes $\hat{c} \in C$, given the lexicon-based instruction. These style-lexicons, are the only source of target-style supervision provided to the LLM, enabling it to make stylistic predictions using parametric knowledge that was acquired during pre-training (Raffel et al., 2020; Brown et al., 2020a), and elicited using lexicon-based instructions.

Meta-Tuning on Source Styles. In order to guide models to draw upon latent lexical knowledge to predict target styles, we meta-tune LLMs (Zhong et al., 2021) to learn to understand style-lexicon relations. During preliminary experiments, we found that it is important to make use of *class randomization* (§2.3) during meta-tuning, e.g. using multiple random words (e.g., “venture”, “quagmire”) to replace the more meaningful style label (e.g., “humorous”), to prevent models from simply memorizing the (source) styles used for fine-tuning. Without randomizing labels, memorization prevents the model from effectively generalizing to interpret lexicons for new styles. In §3.2, we conduct analysis into the impact of randomization by comparing different types of randomization.

Zero-Shot Evaluation on Unseen Target Styles. To make predictions, we provide the model with the target-style lexicon and use rank classification (Sanh et al., 2021), in which we compute the likelihood of each style label, and then pick the one with the highest likelihood as the final prediction.

2.2 A Benchmark for Style Generalization

Style Datasets. We include thirteen language styles that have sentence-level annotated datasets available, covering a wide range of domains, as summarized in Table 1. These come from a variety of sources, including the XSLUE benchmark (Kang and Hovy, 2021), Subjectivity (Pang and Lee, 2004), Shakespeare (Xu et al., 2012) and Readability (Arase et al., 2022) (more details in

Style Dataset	$\|C\|$	**B?**	**Domain**	**#Tra, Val, Test**	**Lexicon Sources**
Age* (Kang and Hovy, 2021)	2	✗	caption	14k, 2k, 2k	ChatGPT, Dict
Country (Kang and Hovy, 2021)	2	✗	caption	33k, 4k, 4k	ChatGPT, Dict
Formality (Rao and Tetreault, 2018)	2	✓	web	209k,10k,5k	NLP (Wang et al., 2010), Dict
Hate/Offense (Davidson et al., 2017)	3	✗	Twitter	22k,1k,1k	NLP (Ahn, 2005), Dict
Humor (CrowdTruth, 2016)	2	✓	web	40k,2k,2k	ChatGPT, Dict
Politeness (Danescu-Niculescu-Mizil et al., 2013)	2	✓	web	10k,0.5k,0.6k	NLP (Danescu-Niculescu-Mizil et al., 2013), Dict
Politics (Kang and Hovy, 2021)	3	✗	caption	33k, 4k, 4k	NLP (Sim et al., 2013), Dict
Readability (Arase et al., 2022)	2	✗	web, Wiki	7k,1k,1k	NLP (Maddela and Xu, 2018), Dict
Romance (Kang and Hovy, 2021)	2	✓	web	2k,0.1k,0.1k	ChatGPT, Dict
Sarcasm (Khodak et al., 2018)	2	✓	Reddit	11k,3k,3k	ChatGPT, Dict
Sentiment (Socher et al., 2013)	2	✗	web	236k,1k,2k	NLP (Mohammad, 2021), Dict
Shakespeare (Xu et al., 2012)	2	✓	web	32k,2k,2k	NLP (Xu et al., 2012), Dict
Subjectivity (Pang and Lee, 2004)	2	✓	web	6k,1k,2k	NLP (Wilson et al., 2005), Dict

Table 1: Statistics of datasets and lexicons. "$|C|$" denotes the number of classes in each style dataset. "B?" indicates whether or not the class distribution is balanced. "#Tra, Val, Test" lists the number of examples in train, validation and test sets. To better compare across different styles, we mapped the original eight classes (i.e., *Under12*, *12-17*, *18-24*, *25-34*, *35-44*, *45-54*, *55-74*, *75YearsOrOlder*) in *Age* dataset into two new classes (i.e., *youthful*, *mature*).

Appendix A). In the cross-style zero-shot setting, a model is fine-tuned on a set of training styles, then evaluated on a separate set of held-out styles with no overlap. For each training style, its training set is used for fine-tuning, and the validation set is used for model selection (Chen and Ritter, 2021). We ensure evaluation style datasets do not share any examples with the training styles. Given space limitations, we present results for one split, which includes Sentiment, Formality, Politeness, Hate/Offense, Readability, Politics, and Subjectivity in the training split, while the remaining six styles are included in evaluation split. Experiments on more style splits are shown in Appendix E.4.

Lexicon Collection. We use stylistic lexicons that have been created by other NLP researchers where possible (listed as "NLP" in Table 1). These lexicons were either manually annotated (Maddela and Xu, 2018) or automatically induced using corpus-based approaches (Danescu-Niculescu-Mizil et al., 2013; Socher et al., 2013). For styles where such lexicons are not readily available, we explore three methods to create lexicons: (i) prompting ChatGPT to generate words for each class of a style, e.g., the words for the "humorous" class are "funny, laugh-out-loud, silly"; (ii) extracting the definition of each class from Google Dictionary,[1] e.g., "being comical, amusing, witty" for the "humorous" class; (iii) having a native English speaker to write a list of words for each style. Creation details and more lexicon examples are provided in Appendix B.

2.3 Lexicon-based Instruction Variations with Class Randomization

To better understand how lexicon-based instructions affect the zero-shot learning abilities of the meta-tuned models (Style-*, e.g., Style-T5), we study variants based on: (i) whether the prompt template contains natural **lang**uage instructions; (ii) the degree of class name **r**andomization. All prompt variants are summarized in Table 2, while example prompts for each variant are shown in Figure 5 in the Appendix. "R#" represents randomizing class names with numerical indices, and "Rw" means using random words as class names in the instruction. We simply use the default English word list in Ubuntu[2] for this randomization. "Rw" uses a much larger set ("vocab size") for higher randomization compared to other variants, which reduces the chance of assigning the same word to the same class in different examples. This class randomization has pros and cons. On one hand, it may hurt performance because it prevents the model from inferring the meaning of classes from class names. On the other hand, it could enhance performance by encouraging the model to genuinely learn the input-class mappings and make use of lexicons, rather than memorizing class names from training styles that are observed during meta-training. In §3, we find class randomization is helpful, possibly because the latter factor outweighs the former (Figure 2).

2.4 Experimental Settings

To assess the effectiveness and generality of lexicon-based instructions, we compare it with

[1] An online service licensed from Oxford University Press: `https://www.google.com/search?q=Dictionary`

[2] `/usr/share/dict/words`

other prompting methods in two learning settings.

2.4.1 Zero-Shot

A model is prompted to predict the evaluation styles without any labeled data. In this setting, We evaluate our Style-* models that are instruction-tuned on the training styles (introduced formally in §2.1). We also experiment with models fine-tuned on general instruction tuning data, including Flan-T5 and LLaMA-2-Chat. For each model, we compare the **Standard** instructions and lexicon-based instructions (i.e., **+ Lex**) without demonstrations (i.e., example sentences for a evaluation style). Both methods utilize the same instruction template which is described in Appendix E.1, except that class names instead of lexicons are used in standard instructions. To construct a lexicon-based instruction, for each class (e.g., "polite" or "impolite") of the style (e.g., politeness), we randomly select m words from the corresponding lexicon, then incorporate them into the instruction. We use $m = 5$ in the main paper and perform an analysis on varied values of m in Appendix E.3.

2.4.2 Few-Shot

We also investigate how different prompting methods perform in the few-shot setting, where a few training examples of the evaluation styles are available. These experiments are not necessarily intended to improve upon the state-of-the-art on this benchmark, but rather to compare the impacts of using in-context examples versus lexicons in enhancing few-shot generalization capabilities.

MetaICL (Min et al., 2022a; Chung et al., 2022). We adapt MetaICL which was developed for meta in-context learning on multiple tasks. During each iteration of fine-tuning, one source style is sampled, and K labeled examples are randomly selected from the train set of that style. Each prompt consists of K demonstrations followed by an input sentence for the model to predict the class. At inference time, the prompt is built similarly to the fine-tuning stage, except that the K demonstrations are sampled from the train set of target styles instead of source styles. Recently, Min et al. (2022b) have shown that ground-truth labels are not required in MetaICL. We re-examine this finding in our task, by experimenting with random and gold example-label mappings in demonstrations. We follow Min et al. (2022b) to set $K = 4$ and $K = 16$.

	no rand.	rand. indices	rand. words	
vocab size	—	3	3	18,843
w/o language	minimal	R#	Rw-	Rw
w/ language	Lang	Lang, R#	Lang,Rw-	Lang,Rw

Table 2: Lexicon-based instruction variants. "vocab" is the fixed set of indices or words, from which a class name can be randomly selected.

MetaICL+Lex. For a more comprehensive comparison between the two sources of supervision (i.e., demonstrations vs. lexicons), we also modify MetaICL to incorporate lexicon signals. Specifically, we concatenate the name of each class with its corresponding lexicon words, and prepend this information to each labeled example to form a modified demonstration. Each prompt contains K modified demonstrations followed by an input sentence.

3 Results and Analysis

We report macro-average F1 for style classification tasks following the XSLUE benchmark (Kang and Hovy, 2021). Our experimental results show that lexicon-based instructions can improve the zero-shot style classification performance in all settings, especially when source style meta-tuning and class randomization are involved.

3.1 Pre-trained Language Models

We experiment with the models T5 (Raffel et al., 2020), GPT-J (Wang and Komatsuzaki, 2021)[3], and LLaMA-2 (Touvron et al., 2023). We also include experiments with the instruction-tuned models Flan-T5 (Chung et al., 2022) and LLaMA-2-Chat (Touvron et al., 2023), as these models have demonstrated the ability to effectively respond to instructions and generalize well to unseen tasks (Chung et al., 2022; Touvron et al., 2023).[4] Implementation details are described in Appendix D.

3.2 Zero-shot Learning Results

Table 3 shows the zero-shot learning results for different methods on various models.

[3] In preliminary studies, we compare the performance of fully fine-tuned, partially (only the last two layers) fine-tuned, and parameter-efficiently fine-tuned GPT-J. Fine-tuning GPT-J with LoRA (Hu et al., 2021) performs the best, so we use it.

[4] We ensure none of the evaluation style datasets appear in the fine-tuning tasks of Flan-T5. However, it remains unclear whether LLaMA-2-Chat has been previously trained on the evaluation styles.

Model	Meta-Tuned?	Instruction	Shakespeare	Romance	Humor	Country	Sarcasm	Age	Avg.
Flan-T5$_{base}$	✗	Standard	33.36	33.33	33.33	43.15	33.33	33.92	35.07
	✗	+ Lex	49.95	51.30	48.66	35.34	49.40	49.02	**47.28**
Style-T5$_{base}$	✓	Standard	33.31	43.57	36.43	19.86	33.37	35.75	33.72
	✓	+ Lex	55.10	78.98	60.56	49.09	49.25	50.80	**57.30**
Style-GPT-J	✓	Standard	58.16	87.82	33.33	53.11	44.10	35.25	51.96
	✓	+ Lex	56.76	83.99	55.86	44.97	48.84	47.47	**56.32**
LLaMA-2-Chat (7B)	✗	Standard	60.20	85.72	43.84	49.19	36.02	38.91	52.31
	✗	+ Lex	62.59	88.95	51.01	50.88	42.88	36.54	**55.47**
LLaMA-2-Chat (13B)	✗	Standard	61.99	97.00	47.42	17.96	43.26	48.16	52.63
	✗	+ Lex	63.49	95.00	55.15	24.41	44.66	53.88	**56.10**
LLaMA-2 (7B)	✗	Standard	42.13	64.41	37.38	48.27	48.84	37.13	46.36
	✗	+ Lex	50.21	77.86	45.44	49.86	47.72	47.63	**53.12**
Style-LLaMA (7B)	✓	Standard	40.91	41.65	48.88	48.92	49.02	49.80	46.53
	✓	+ Lex	59.03	88.97	57.64	51.52	50.83	50.53	**59.75**

Table 3: Zero-shot performance on the unseen evaluation styles. We compare the models fine-tuned on general instruction tuning data (i.e., not meta-tuned) and the "Style-*" models that are instruction-tuned on our training styles (i.e., meta-tuned). For each model, we evaluate its zero-shot learning capabilities when the standard and lexicon-based instructions are used, respectively.

Baseline Method	Shakespeare	Romance	Humor	Country	Sarcasm	Age	Avg.
Majority Classifier	33.30	33.30	33.30	49.20	33.30	35.30	36.28
Lex Frequency	$59.91_{83\%}$	$32.89_{28\%}$	$33.33_{0.49\%}$	$50.79_{5.7\%}$	$33.33_{0.59\%}$	$37.85_{18\%}$	41.35
Lex Emb Sim (Word2Vec)	49.06	33.33	33.54	49.30	33.33	50.84	41.57
Lex Emb Sim (SentenceBert)	52.00	69.81	57.62	31.12	47.91	49.96	51.40

Table 4: Performance of zero-shot baselines. We compare three approaches: (1) The majority classifier, which predicts the majority label in training data. (2) The lexicon frequency baseline, which counts the occurrence of words from an input sentence in each class's lexicon and then predicts the class with the highest count; the subscript on the score reflects the lexicon usage, i.e., the percentage (%) of evaluation data that contains at least one word from the corresponding lexicons. (3) The lexicon embedding similarity method, which calculates the cosine similarity between the embeddings of lexicon words for each class and an input, predicting the class with the highest similarity.

Lexicon-based instructions outperform the standard instructions. In the zero-shot setup, incorporating lexicons into instructions demonstrates a significant advantage over the standard instructions without lexicon information across all the experimented models. For example, after integrating lexicons into instructions and randomizing classes, + Lex improves upon the standard instructions by an average of 23.58 F1 points on Style-T5 and an average of 13.22 F1 points on Style-LLaMA. One possible explanation for this improvement is that, if we fix the class names during source fine-tuning, the model tends to memorize these names instead of learning from other signals. This is not ideal as our goal is to predict unseen styles and thus learning from the lexicon is important. By randomizing the class names during instruction tuning, the model is able to focus more on lexicons and other common information shared across styles (e.g., instructions) rather than style-specific tokens (e.g., class names). This suggests that format transfer, i.e., classification based on the relevance between each style lexicon and the input sentence, is crucial. More experiments on randomization are shown in §3.2.

While LLaMA-2-Chat models exhibit impressive performance using standard instructions, by simply integrating lexicons into instructions, their performance can be further enhanced in most styles. Notably, F1 improves from 43.84 to 51.01 for Humor style on LLaMA-2-Chat (7B).

Instruction tuning on training styles with lexicons enhances the zero-shot performance on evaluation styles, compared to fine-tuning with general instructions. Both Style-T5 and Style-LLaMA demonstrate a significant performance improvement upon their general instruction-tuned counterparts, i.e., Flan-T5 and LLaMA-2-Chat, when lexicon is included in instructions. For in-

	Method	Shakespeare	Romance	Humor	Country	Sarcasm	Age	Avg.
Examples w/ *random* labels	MetaICL$_4$	44.37$_{\pm 6.99}$	56.21$_{\pm 26.64}$	37.82$_{\pm 5.02}$	41.84$_{\pm 18.46}$	35.55$_{\pm 2.94}$	40.96$_{\pm 11.19}$	42.79
	MetaICL$_4$+Lex	39.80$_{\pm 1.47}$	64.58$_{\pm 18.72}$	38.59$_{\pm 4.41}$	49.72$_{\pm 0.44}$	43.77$_{\pm 6.52}$	35.30$_{\pm 0.00}$	45.29
	MetaICL$_{16}$	55.49$_{\pm 11.66}$	66.91$_{\pm 20.48}$	36.11$_{\pm 4.58}$	7.74$_{\pm 4.67}$	33.33$_{\pm 0.00}$	31.24$_{\pm 0.00}$	38.47
Examples w/ *gold* labels	MetaICL$_4$	64.30$_{\pm 13.01}$	53.53$_{\pm 27.30}$	49.79$_{\pm 12.46}$	49.29$_{\pm 0.01}$	34.28$_{\pm 1.57}$	36.21$_{\pm 1.25}$	47.90
	MetaICL$_4$+Lex	43.90$_{\pm 8.06}$	75.80$_{\pm 6.52}$	42.78$_{\pm 3.99}$	49.42$_{\pm 0.36}$	38.62$_{\pm 3.69}$	35.30$_{\pm 0.00}$	47.63
	MetaICL$_{16}$	72.93$_{\pm 8.15}$	95.79$_{\pm 0.84}$	52.05$_{\pm 8.52}$	47.90$_{\pm 3.07}$	33.33$_{\pm 0.00}$	35.30$_{\pm 0.00}$	56.22

Table 5: Few-shot learning of GPT-J. The subscript of MetaICL represents the number (K) of demonstrations in one prompt. For each method (MetaICL$_K$, or MetaICL$_K$+Lex), we choose a set of K examples with five different random seeds. More results on varying values of K are shown in Appendix E.6. We also modify lexicon-based instructions for few-shot learning and compare it with other few-shot learning methods in Appendix E.5.

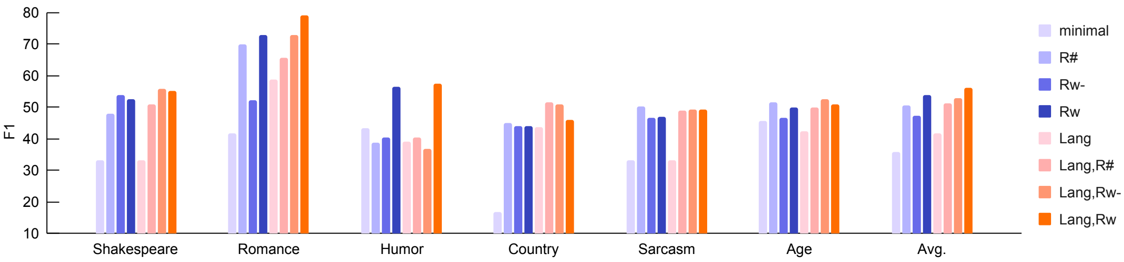


Figure 2: Zero-shot performance when fine-tuning with different lexicon-based instruction variants. Instruction tuning with class Randomization shows advantages over those without. Instructions with natural language perform generally better than those without.

stance, Style-LLaMA (7B) outperforms LLaMA-2-Chat (7B) in five out of six styles, achieving an average increase of 4.28 F1 points. This suggests the benefits of lexicon-based instructions and the effectiveness of instruction tuning on training styles.

Class randomization matters in lexicon-based prompting. We study the impact of natural language descriptions and class randomization in our approach by independently fine-tuning Style-T5 on the training styles using the eight variants (§2.3) listed in Table 2. Our experimental results in Figure 2 show that introducing class randomization can improve the zero-shot performance on the six unseen evaluation styles consistently. For example, the average F1 improves from 35.58 (minimal) to 50.54 (R#).

3.3 Few-shot Learning Results

Table 5 shows results of few-shot learning methods.

Incorporating lexicons in few-shot learning reduces the sensitivity to example selection. Different choices of the examples selected for few-shot learning can lead to highly different performance (Zhao et al., 2021; Liu et al., 2022). Hence how to reduce the sensitivity due to example selection has become an important research question. It is observed that by introducing lexicon into prompts, the standard deviation of performance across five runs generally decreases. For example, MetaICL$_4$ performs extremely unreliably on Romance with a high standard deviation of 27.30, while MetaICL$_4$+Lex not only improves performance but also stabilizes inference with a standard deviation dropped to 6.52. This may suggest that using lexicons can reduce a model's dependence on the selected few-shot examples (Liu et al., 2022).

Introducing lexicons into in-context examples can be beneficial when gold labels are not available. When the examples of the evaluation style are randomly labeled, introducing lexicon into MetaICL is generally more useful than increasing the number of examples. For example, MetaICL$_{16}$ falls short of MetaICL$_4$ by an average of 4.32 F1 points over the six styles, whereas MetaICL$_4$+Lex shows an improvement over MetaICL$_4$, increasing the average score by 2.5 points. When ground-truth labels are accessible, MetaICL$_{16}$ showcases a superior average performance, suggesting that increasing the number of demonstration might be more effective in this case.

4 Generalization to Novel Styles

In prior sections, we established the effectiveness of our method on established NLP style datasets. To demonstrate that our method, which fine-tunes models to interpret lexicon-based instructions, is able to generalize beyond styles that have been previously studied in the NLP community, we next use LLMs to semi-automatically propose new styles, and then generate instances of text presenting each style (i.e., labeled examples). The new styles generated in this section are then used to evaluate models' capability to generalize to styles that include but are not limited to niche literary genres, or rapidly evolving communication styles in social media (see examples in Appendix Table 15).

4.1 A Diverse Collection of New Styles

Style Creation. We compiled a diverse collection of language styles by initiating the data generation based on the thirteen styles listed in Table 1. This initial set served as a seed for prompting LLaMA-2-Chat 70B to generate different style classification tasks using in-context examples. We filtered out any tasks that did not align with our textual classification objective. To encourage diversity, a new task is added to the pool only when its ROUGE-L similarity with any existing task is less than 0.6. This process produced 58 new unique style classification tasks. We then randomly divided these tasks into the training and evaluation split, avoiding task overlap. To further enrich the diversity, we developed and added 5 additional tasks to the evaluation split, such as composite chatbot styles (e.g., characterized by a blend of empathetic, colloquial, and humorous responses), and writing styles of various authors. Please refer to Appendix C.1 for additional details on the style creation process. The full list of 63 tasks generated for our study can be found in Appendix Table 14.

Lexicon Creation. Depending on the construction method, these lexicons may vary in quality and size from a few words to thousands. Nevertheless, we will show the benefits of our method with as few as five words per style sampled from lexicons (Appendix E.3). Our ablation studies (see Appendix E.2) demonstrate the robustness of lexicon-based instructions across various lexicon creation methods, particularly when class randomization is applied. Hence for each new style, we prompted LLaMA-2-Chat 70B (as detailed in Appendix C.2) to generate a concise lexicon for each style class, comprising up to five words or phrases.

Labeled Example Generation. We employed LLaMA-2-Chat 7B to generate 100 unique examples for each class in our training style split, which results in a training style dataset $\mathcal{D}_{\text{train}}$. For the evaluation style dataset $\mathcal{D}_{\text{eval}}$, we leveraged GPT-4 (OpenAI, 2023) to create high-quality stylistic examples. Through the OpenAI API, we generated 20 examples for each class at a total cost of \$9.11. To assess the quality of $\mathcal{D}_{\text{eval}}$, we asked three human annotators[5] to review the labeled examples generated by GPT-4. Details about this process are presented in Appendix C.3.

Statistics. Our data generation process produced a collection of 11,358 distinctive examples, spanning 63 varied style classification tasks. Table 6 describes the statistics of our data. The distribution of K-class tasks (where K is the number of distinct style classes to be distinguished) is illustrated in Figure 4, showcasing the diverse range of styles included in our analysis. Examples of the generated style data and lexicons are shown in Appendix C.4.

statistics	$\mathcal{D}_{\text{train}}$	$\mathcal{D}_{\text{eval}}$
# of classification tasks	43	20
# of examples	10,308	1,050
avg. # of classes per example	3.20	3.12
avg. example length (in words)	30.47	21.40
avg. lexicon size (in words/phrases)	4.11	3.74

Table 6: Statistics of model-generated datasets.

Inter-Rater Agreement on Evaluation Set. To measure the reliability of $\mathcal{D}_{\text{eval}}$, we compute inter-annotator agreement (Krippendorff's alpha) over a shared set of 500 randomly selected annotation examples. Annotators were instructed to assess the accuracy of labels for examples generated by GPT-4 and make necessary corrections. Each example was independently reviewed by three annotators. The score of 93.27% reflects substantial agreement.

4.2 Experiments

Experiment Setup. We evaluated the zero-shot performance of LLaMA-2-Chat (7B, 13B) and Style-LLaMA (7B) on $\mathcal{D}_{\text{eval}}$. Given the balanced class distribution in this set, we report accuracy in Table 7. We also included Style-LLaMA+ (7B), which fine-tuned the LLaMA-2 model on a mix of benchmark training styles and the training set $\mathcal{D}_{\text{train}}$ generated by LLaMA-2-Chat 7B. It is important to note that the training set $\mathcal{D}_{\text{train}}$ and the evaluation set $\mathcal{D}_{\text{eval}}$ were created by different language models, ensuring that there is no overlap in styles or data. Implementation details are described

[5]The three annotators include: one of the authors, a graduate student in CS, and a mathematician.

	Standard	+ Lex (ours)
Random Classifier	36.65	
LLaMA-2-Chat (7B)	53.09	56.23
Style-LLaMA (7B)	46.25	58.71
Style-LLaMA+ (7B)	65.46	**74.31**
LLaMA-2-Chat (13B)	56.80	59.75

Table 7: Zero-shot learning on $\mathcal{D}_{\text{eval}}$. Lexicon-based instructions improve the zero-shot generalization capabilities of the studied models.

in Appendix D. A baseline was set by randomly assigning a class to each example, averaging the results over five different seeds.

Results Table 7 demonstrates the advantages of lexicon-based instructions over the standard instructions. Notably, Style-LLaMA and Style-LLaMA+ show the most significant performance gains, with an average improvement of 12.46 and 8.85, respectively. This is likely because lexicon-based instruction-tuning enhances their adaptability to new styles through more effective lexicon usage. Furthermore, Style-LLaMA+ shows a substantial improvement over other models, suggesting that the inclusion of a diverse set of model-generated style training data can effectively enhance the performance. The peak score of Style-LLaMA+ with lexicon integration suggests that the combination of additional training data and lexicon-based instructions might be the most effective approach for generalization among the evaluated methods.

5 Related Work

Style classification. Research in NLP has studied various language styles. Kang and Hovy (2021) provided a benchmark for fully-supervised style classification that combines many existing datasets for style classification, such as formality (Rao and Tetreault, 2018), sarcasm (Khodak et al., 2018), Hate/Offense (i.e., toxicity) (Davidson et al., 2017), politeness (Danescu-Niculescu-Mizil et al., 2013), and sentiment (Socher et al., 2013; Wang et al., 2021). Other writing styles include but are not limited to readability (i.e., simplicity) (Arase et al., 2022), Shakespearean English (Xu et al., 2012), subjectivity (Pang and Lee, 2004), biasedness (Pryzant et al., 2020) and engagingness (Jin et al., 2020). Despite an extensive range of style classification tasks studied in prior research, zero-shot or cross-style classification is relatively underexplored (Puri and Catanzaro, 2019). In particular, much of the cross-style research thus far has focused on text generation tasks (Jin et al., 2022; Zhou et al., 2023), rather than classification. In this study, we aim to address this gap in the literature by concentrating on zero-shot style classification across a collection of diverse styles.

Language model prompting. Large language models (LLMs), such as GPT-3 (Brown et al., 2020b), demonstrate impressive zero-shot learning abilities by conditioning on an appropriate textual context, i.e., prompts, or natural language instructions. Since then, how to design appropriate prompts has become a popular line of research (Schick and Schütze, 2021; Sanh et al., 2021; Chung et al., 2022). In this work, we propose to incorporate lexicons into instructions and teach the model to better utilize stylistic lexicon knowledge through instruction tuning. Recently, Zhou et al. (2023) specified the styles in instructions as constraints to improve controlled text generation. Parallel to our study, Gao et al. (2023) investigated label descriptions to enhance zero-shot learning for topic and sentiment classification. We focus on style classification, a challenging area in NLP characterized by its extensive scope and complexity, encompassing a wide range of stylistic expression across various domains of text. In order to improve the generalization capabilities of instruction-tuned models, we replace class names in instructions with entirely random words during fine-tuning on training styles. This is similar to Zhao et al. (2022), which indexes and shuffles slot descriptions in prompts used for dialogue state tracking. Moreover, our work differs from the standard practice in previous studies (Min et al., 2022b; Zhao et al., 2022; Wei et al., 2023), where a pre-defined set of class names, is equal in size to the number of labels in the associated datasets.

6 Conclusion & Discussion

In this work, we study zero-shot style classification using large language models in combination with lexicon-based instructions. Experiments show that conventional instructions often struggle to generalize across diverse styles. However, our lexicon-based instruction approach demonstrates the potential to fine-tune models for improved zero-shot generalization to unseen styles. Our method may generalize to generation tasks (e.g., cross-style transfer), which we would like to explore in future work.

Limitations

In our method, we leverage the lexicons we have collected (as detailed in Table 1). However, it is important to acknowledge that a potential limitation of our approach lies in the possibility of different performance outcomes when using lexicons of varying qualities. While we have conducted comparisons between lexicons from different sources in Appendix E.2, it is plausible that utilizing different lexicons could yield different results. Another limitation is that we only include a limited set of styles in English for evaluation due to availability of high-quality style datasets and lexicons. We leave data curation and evaluation for additional styles and languages to future work.

Ethical Considerations

Style classification is widely studied in the NLP research community. We strictly limit to using only the existing and commonly used datasets that are related to demographic information in our experiments. As a proof of concept, this research study was only conducted on English data, where human annotations for multiple styles are available for use in the evaluation. We also acknowledge that linguistic styles are not limited to what are included in this paper, and can be much more diverse. Future efforts in the NLP community could further extend research on stylistics to more languages and styles.

References

Luis Von Ahn. 2005. Useful resources: Offensive/profane word list. https://www.cs.cmu.edu/~biglou/resources/.

Yuki Arase, Satoru Uchida, and Tomoyuki Kajiwara. 2022. CEFR-based sentence difficulty annotation and assessment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6206–6219, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020b. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yang Chen and Alan Ritter. 2021. Model selection for cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

CrowdTruth. 2016. Short Text Corpus For Humor Detection. http://github.com/CrowdTruth/Short-Text-Corpus-For-Humor-Detection. [Online; accessed 1-Oct-2019].

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 250–259, Sofia, Bulgaria. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Tim Dettmers and Luke Zettlemoyer. 2023. The case for 4-bit precision: k-bit inference scaling laws. In *International Conference on Machine Learning*, pages 7750–7774. PMLR.

Peter Sheridan Dodds, Eric M Clark, Suma Desu, Morgan R Frank, Andrew J Reagan, Jake Ryland Williams, Lewis Mitchell, Kameron Decker Harris, Isabel M Kloumann, James P Bagrow, et al. 2015. Human language reveals a universal positivity bias. *Proceedings of the national academy of sciences*, 112(8):2389–2394.

Jacob Eisenstein. 2017. Unsupervised learning for lexicon-based classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Lingyu Gao, Debanjan Ghosh, and Kevin Gimpel. 2023. The benefits of label-description training for zero-shot text classification. *arXiv preprint arXiv:2305.02239*.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics*, 48(1):155–205.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, Lisa Orii, and Peter Szolovits. 2020. Hooks in the headline: Learning to generate headlines with controlled styles. *arXiv preprint arXiv:2004.01980*.

Dongyeop Kang and Eduard Hovy. 2021. Style is not a single variable: Case studies for cross-style language understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Junmo Kang, Hongyin Luo, Yada Zhu, James Glass, David Cox, Alan Ritter, Rogerio Feris, and Leonid Karlinsky. 2023. Self-specialization: Uncovering latent expertise within large language models. *arXiv preprint arXiv:2310.00160*.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Mounica Maddela and Wei Xu. 2018. A word-complexity lexicon and a neural readability ranking model for lexical simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3749–3760, Brussels, Belgium. Association for Computational Linguistics.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022a. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*.

Saif Mohammad and Peter Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*.

Saif M. Mohammad. 2021. Sentiment analysis: Automatically detecting valence, emotions, and other affectual states from text. In Herb Meiselman, editor, *Emotion Measurement (Second Edition)*. Elsevier.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Annual Meeting of the Association for Computational Linguistics*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):480–489.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. *CoRR*, abs/2110.08207.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Yanchuan Sim, Brice D. L. Acree, Justin H. Gross, and Noah A. Smith. 2013. Measuring ideological proportions in political speeches. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 91–101, Seattle, Washington, USA. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*.

Yla R Tausczik and James W Pennebaker. 2010. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*.

Zhiyang Teng, Duy-Tin Vo, and Yue Zhang. 2016. Context-sensitive lexicon features for neural sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1629–1638, Austin, Texas. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Gaurav Verma and Balaji Vasan Srinivasan. 2019. A lexical, syntactic, and semantic perspective for understanding style in text. *arXiv preprint arXiv:1909.08349*.

Rob Voigt, Nicholas P Camp, Vinodkumar Prabhakaran, William L Hamilton, Rebecca C Hetey, Camilla M Griffiths, David Jurgens, Dan Jurafsky, and Jennifer L Eberhardt. 2017. Language from police body camera footage shows racial disparities in officer respect. *Proceedings of the National Academy of Sciences*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Shuai Wang, Guangyi Lv, Sahisnu Mazumder, and Bing Liu. 2021. Detecting domain polarity-changes of words in a sentiment lexicon. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3657–3668.

Tong Wang, Julian Brooke, and Graeme Hirst. 2010. Inducing lexicons of formality from corpora.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wei Xu. 2017. From shakespeare to Twitter: What are language styles all about? In *Proceedings of the Workshop on Stylistic Variation*, pages 1–9, Copenhagen, Denmark. Association for Computational Linguistics.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *COLING*, pages 2899–2914.

Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. Description-driven task-oriented dialog modeling.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *ArXiv*, abs/2102.09690.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2856–2878.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023. Controlled text generation with natural language instructions. *arXiv preprint arXiv:2304.14293*.

A Benchmark Datasets Details

The XSLUE benchmark, designed for exploring cross-style language understanding, encompasses 15 styles (Kang and Hovy, 2021). We choose 10 writing styles from XSLUE based on their suitability for our task. Specifically, we consider the task type (i.e., whether the task is classification or not), task granularity (e.g., whether the annotated style is sentence-level or not), expressiveness at both the word and phrase level (i.e., the possibility of expressing a style with lexicons). For example, the TroFi dataset for style Metaphor is not used because it is focused on the literal usage of one specific verb in a sentence. Take the verb "drink" as an example, it is a literal expression in the sentence "'I stayed home and drank for two years after that,' he notes sadly", whereas in "So the kids gave Mom a watch, said a couple of nice things, and drank a retirement toast in her honor", "drink" is non-literal.

B Benchmark Lexicons Details

B.1 Lexicon Creation

ChatGPT-generated Lexicons. Prior work has used models, such as BERT to generate class vocabularies for topic classification (Meng et al., 2020). Inspired by this approach, we utilize the knowledge of LLMs by prompting them to generate a list of words that express the specific class of a style. In a preliminary study, we experimented with many LLMs, including BERT, GPT-J, GPT-NeoX, GPT-3.5 and ChatGPT. Among all, ChatGPT performs the best, so we use it to generate the lexicons. Table 8 shows the prompts we used for ChatGPT. Figure 3 presents some examples of ChatGPT output.

Dictionary-based Lexicons. We also considered lexicons generated by extracting the definition of each style from Google Dictionary.

B.2 Statistics and Examples of Lexicons

Table 9 provides the statistics of NLP and ChatGPT lexicons used in the experiments. Table 10 shows examples of lexicons from different sources.

C Model-Generated Data For Generalization Experiments

Recall in §4 that in order to further evaluate the generalization capabilities of our porposed approach, we collected a diverse collection of styles using LLMs. Here we provide more details throughout the data generation process, including style creation (§C.1), lexicon generation (§C.2), and labeled example (i.e., instance) generation (§C.3).

C.1 Style Creation

We initiated the process of style classification task generation based on the thirteen styles outlined in our benchmark (refer to Table 1). We had one author write the style classification instruction for each of these thirteen styles. During the task generation process, we randomly selected eight in-context examples from our pool, including three seed tasks and five model-generated tasks. We employed LLaMA-2-Chat 70B for new task generation. The template used for prompting these new style classification tasks are detailed in Table 11. To ensure the diversity of the generated style classification tasks, a new task is added to the pool only when its ROUGE-L similarity with any existing task is less than 0.6. This process resulted in a total of 58 model-generated tasks, which we divided into 43 training tasks and 15 evaluation tasks. In order to further enrich the diversity of the evaluation task split, we designed 5 additional style classification tasks and incorporated them into the evaluation task split. Overall, this data generation process produces a total of 43 training style classification tasks and 20 evaluation style classification tasks. We present the full list of 63 generated style classification tasks in Table 14.

C.2 Lexicon Creation

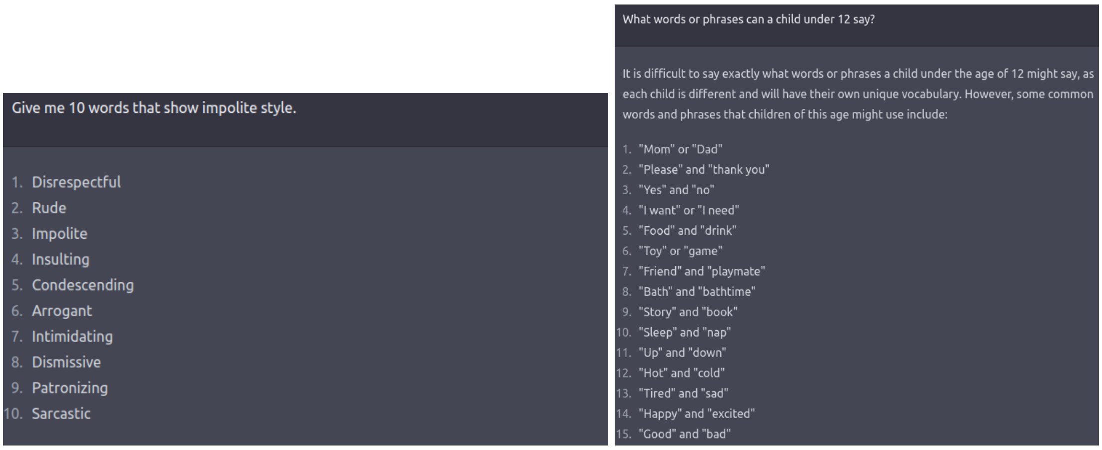
After creating the training and evaluation tasks, we employed LLaMA-2-Chat 70B to generate a concise lexicon for each class in the style classification tasks, using in-context examples. Our ablation studies, as detailed in §E.3, revealed that a lexicon consisting of just five words or phrases are sufficient for effective generalization to new styles. So we restricted the lexicon size for each style class to five words or phrases. The template used for prompting the generation of style class lexicons are displayed in Table 12.

C.3 Labeled Example Generation

We prompted LLaMA-2-Chat 7B to generate labeled examples for our training style classification tasks, and GPT-4 to generate examples for our evaluation tasks. Both utilize the same prompting template presented in Table 13 for labeled example generation.

Style	Class	ChatGPT Prompt
Politeness	impolite	Give me 10 words that show impolite style. Give me 20 words or short phrases that people may use when they show impolite attitude towards others.
Romance	literal	What's the difference between literal text and romantic text? Give me 20 words or short phrases that show the literal style rather than romantic style.
Humor	humorous	Give me 10 words that show humorous style. Give me 20 words or short phrases that people may use in text to show humor.
	literal	What's the difference between literal text and humorous text? Give me 20 words or short phrases that show the literal style rather than humorous style.
Sarcasm	sarcastic	Give me 10 words that show sarcastic style. Give me 20 words or short phrases that people may use in text to show sarcasm.
	literal	What's the difference between literal text and sarcastic text? Give me 20 words or short phrases that show the literal style rather than sarcastic style.
Age	under12	Give me some words or phrases that an under-12-year-old child might say or write. What words or phrases can a child under 12 say? Imagine that you are 8 years old, what words or phrases do you often use in communication and writing?
	12-17	Give me some words or phrases that people aged 12-17 might say or write. What words or phrases can a teenager aged 12-17 say? Imagine that you are 14 years old, what words or phrases do you often use in communication and writing?
	18-24	Give me some words or phrases that people aged 18-24 might say or write. What words or phrases can a person aged 18-24 say? Imagine that you are 21 years old, what words or phrases do you often use in communication and writing?
	25-34	Give me some words or phrases that people aged 25-34 might say or write. What words or phrases can a person aged 25-34 say? Imagine that you are 30 years old, what words or phrases do you often use in communication and writing?
	35-44	Give me some words or phrases that people aged 35-44 might say or write. What words or phrases can a person aged 35-44 say? Imagine that you are 40 years old, what words or phrases do you often use in communication and writing?
	45-54	Give me some words or phrases that people aged 45-54 might say or write. What words or phrases can a person aged 45-54 say? Imagine that you are 50 years old, what words or phrases do you often use in communication and writing?
	55-74	Give me some words or phrases that people aged 55-74 might say or write. What words or phrases can a person aged 55-74 say? Imagine that you are 65 years old, what words or phrases do you often use in communication and writing?
	75Years OrOlder	Give me some words or phrases that people aged 75 or older might say or write. What words or phrases can a person aged 75 or older say? Imagine that you are 80 years old, what words or phrases do you often use in communication and writing?

Table 8: Prompts used for ChatGPT to generate lexicon. Since we map the *Age* dataset to a binary one, we also map the corresponding lexicons of its original age classes to the new classes. For example, the *youthful* lexicon contains the contents of *Under12*, *12-17* and *18-24* lexicons.



(a) impolite (b) under12

Figure 3: Examples of ChatGPT output for different style classes.

Style	**Class**	**Lex Src**	**Lex Size (# of words/phrases)**
Age	youthful	ChatGPT	98
	mature	ChatGPT	65
Country	U.K	ChatGPT	131
	U.S.A	ChatGPT	127
Formality	formal	NLP	330
	informal	NLP	370
	hate	NLP	178
Hate/Offense	offensive	NLP	1403
	neither	ChatGPT	5
Humor	humorous	ChatGPT	21
	literal	ChatGPT	6
Politeness	polite	NLP	110
	impolite	ChatGPT	54
Politics	LeftWing	NLP	2581
	Centrist	NLP	1231
	RightWing	NLP	2416
Readability	simple	NLP	10290
	complex	NLP	4890
Romance	romantic	ChatGPT	58
	literal	ChatGPT	5
Sarcasm	sarcastic	ChatGPT	34
	literal	ChatGPT	2
Sentiment	positive	NLP	204
	negative	NLP	292
Shakespeare	shakespearean	NLP	1524
	modern	NLP	1524
Subjectivity	subjective	NLP	5569
	objective	NLP	2653

Table 9: Statistics of benchmark style lexicons.

Style	**Class**	**Lex Src**	**Lex**
Formality	formal	NLP	admittedly, albeit, insofar...
		Dict	in accordance with rules of convention or etiquette; official
	informal	NLP	dude, kinda, sorta, repo...
		Dict	having a relaxed, friendly, or unofficial style
Humor	humorous	ChatGPT	funny, laugh-out-loud, silly...
		Dict	being comical, amusing, witty
		human	chuckle, wisecrack, hilarious...
	literal	ChatGPT	grim, formal, solemn, dour...
		Dict	not humorous; serious
		human	analysis, scrutinize, enforce...

Table 10: Examples of lexicons. "Class" represents the category in a style. Each lexicon contains words or phrases that express or describe the class. "Lex Src" indicates how the lexicon is collected (§2.2).

```
Come up with a series of textual classification tasks about writing styles.
Try to specify the possible output labels when possible.

Task 1: {instruction for existing task 1}
Task 2: {instruction for existing task 2}
Task 3: {instruction for existing task 3}
Task 4: {instruction for existing task 4}
Task 5: {instruction for existing task 5}
Task 6: {instruction for existing task 6}
Task 7: {instruction for existing task 7}
Task 8: {instruction for existing task 8}
Task 9:
```

Table 11: Prompt template used for generating new style classification tasks. 8 existing instructions are randomly sampled from the task pool for in-context demonstration. The model is allowed to generate instructions for new tasks, until it stops its generation or reaches its length limit.

```
You are a helpful AI assistant. Generate a few words that describe or exhibit
the target style. If the words cannot fully express the characteristics of the
style, define the style with phrases or short sentences.

Example
Style class 1: {lexicon words/phrases for style class 1}

Example
Style class 2: {lexicon words/phrases for style class 2}

...

Example
Style class 8: {lexicon words/phrases for style class 8}

Example
Style class 9:
```

Table 12: Prompt template used for generating style class lexicon.

```
You are a helpful AI assistant. Given the classification task definition and the possible
output labels, generate an input that corresponds to each of the class labels. Try to generate
high-quality inputs with varying lengths.

Task: Classify the sentiment of a sentence. The possible output labels are: positive,
negative.
Label: positive
Sentence: I had a great day today. The weather was beautiful and I spent time with friends and
family.
Label: negative
Sentence: I was really disappointed by the latest superhero movie.

Task: Categorize the writing style of a given piece of text into romantic, or not romantic.
Label: romantic
Text: A lot of people spend their whole lives looking for true love and ultimately fail. So how
ungrateful would I be, if I let our love fade? That @ Ys how you know, my love is here to stay.
Label: not romantic
Text: I need you to submit this proposal as soon as possible.

...

Task: {instruction for the target task}
```

Table 13: Prompt template used for generating the example for classification tasks.

Style Classification Task	Classes
Identify the type of writing style used in a given text.	narrative, descriptive, expository, persuasive
Determine whether the given text contains any errors in grammar, spelling, or punctuation.	error-free, erroneous
Classify the style of a poem into one of the four types.	sonnet, haiku, free verse, limerick
Categorize the emotion of the utterances.	angry, disgusted, fearful, happy, sad
Determine the level of organization in the text.	well-organized, disorganized
Classify the style of a text according to its structure.	chronological, non-chronological
Classify the text according to its tone.	friendly, hostile, neutral
Define the writing style "Infotainment" as "merging informative writing with an entertaining approach". Define the writing style "Techeative" as "blending technical writing (e.g. precise descriptions of complex subjects) with creative elements to make it more engaging and understandable". Classify the style of a presentation into one of the above two categories.	Infotainment, Techeative
Classify the style of a text according to its content and language use.	rational, irrational
Evaluate the level of clarity in the text.	clear, unclear
Classify the text style according to its tone and language use.	nostalgic, reflective, analytical
Classify the style of a text according to its content and language use.	creative, conventional
Identify the author's voice style in a given text.	authoritative, unreliable
Evaluate the level of emotional appeal in the text.	low emotional appeal, high emotional appeal
Determine the level of originality in a story.	original, somewhat original, not original
Evaluate the level of credibility in the text.	credible, moderately credible, not credible
Read a passage, and select the topic for this passage based on the content and text style.	finance, politics, health, education, technology, entertainment
Read the summary of a book and categorize its genre.	science fiction, romance, thriller, biography
Determine the primary intention behind the author's writing of a specific text.	persuasive, informative, entertaining, educational
Classify the text style according to its tone and language use.	assertive, submissive
Classify the text style according to its tone and language use.	strong, weak
Classify text style.	conversational, academic
Determine the most likely author based on the writing style.	Hemingway, Joyce, Kafka, Hurston, Christie
Classify the text style according to its tone and language use.	monotonous, engaging
Classify the content of a piece of text.	spam, ham
Read the text and classify its style.	fictional, non-fictional
Evaluate the mood of a song based on its lyrics.	relaxing, energizing, romantic, melancholic
Identify the rhetorical devices used in a given text.	onomatopoeia, alliteration, hyperbole, repetition, oxymoron
Classify the text as one of the following: journalistic, academic, or literary.	journalistic, academic, literary
Assess how supportive the context is in response to a request for help.	very supportive, moderately supportive, not supportive
Classify the text according to its tone and language use.	realistic, idealistic

continued on next page

continued from previous page

Style Classification Task	**Classes**
Given a famous quote, classify its tone style into one of the four categories.	inspirational, funny, philosophical, sarcastic
Classify the text according to its tone and language use.	confident, uncertain, timid
Carefully review the provided text and assess its level of rigor.	rigorous, careless
Classify the author's attitude towards the topic.	enthusiastic, uninterested
Assess the difficulty level of academic texts, and choose the label from the following four options.	elementary, intermediate, advanced, expert
Analyze the given text and determine whether it contains any biases.	biased, unbiased
Classify the style of an example.	adventurous, cautious, conservative
Classify the text according to its tone.	optimistic, pessimistic, neutral
Classify the text style.	logical, emotional
Classify the text according to its tone and language style.	apologetic, accusatory, grateful, condescending
Determine the response style by examining the content and the quality of a response.	helpful and harmless, helpful and harmful, helpless and harmless, helpless and harmful
Identify the style of a sonnet by analyzing the rhyme scheme of its first four lines, each separated by a newline symbol.	Shakespearean sonnet, Petrarchan sonnet.
Identify the style of a poetry by analyzing the rhyme scheme of its first four lines, each separated by a newline symbol.	ABAB, AABB
Carefully review the provided text and determine the nature of its writing style.	machine-generated text, human-written text
XXX and YYY are two Ph.D. students who often engage in writing papers. XXX has a penchant for employing a variety of fancy words and clauses in the writing, whereas YYY favors a style that is more concise and straightforward, focusing on brevity and clarity. Given a piece of text, determine who is more likely to be the author based on the writing style.	XXX, YYY
Determine the level of coherence in a piece of writing.	coherent, incoherent
Determine whether the text contains any sensitive information such as personal data, financial information, or explicit content.	sensitive, non-sensitive
Classify text format based on the language style used.	editorial, blog post, research paper, poem, script
Determine if a tweet contains misinformation.	true, misleading
Determine the level of nuance in a piece of writing.	nuanced, somewhat nuanced, not nuanced
Classify text style according to its intended audience.	general public, experts, children, young adults
Analyze the tone of a customer review for a product.	satisfied, dissatisfied, mixed feelings
Determine the tone of the text.	serious, ironic, condescending
Evaluate the level of technical jargon used in the text.	technical, non-technical
Classify the attitude of the author into either ẅanting to helpör p̈erfunctory.̈	helpful, unhelpful
Classify the poetry style type.	ballad, acrostic, ode, elegy, limerick
Define the style of a "empathetic, colloquial, humorous, lively" response as "teddy bear". Define the style of a "calm, caring, professional, earnest" response as "psyduck". Classify the style of responses made by a senior AI Assistant.	teddy bear, psyduck
Analyze the content and language style of the support ticket or email and classify its urgency level.	high urgency, medium urgency, low urgency, informational
Given a sentence, detect if there is any potential stereotype in it.	stereotyped, non-stereotyped
Determine the level of conciseness in a piece of writing.	concise, verbose
A desirable trait in a human-facing dialogue agent is to appropriately respond to a conversation partner that is describing personal experiences, by understanding and acknowledging any implied feelings - a skill we refer to as empathetic responding. Classify the response style.	empathetic, indifferent
Identify the rhetorical devices used in a given text.	metaphor, simile, personification

Table 14: 63 generated style classification tasks in §4.

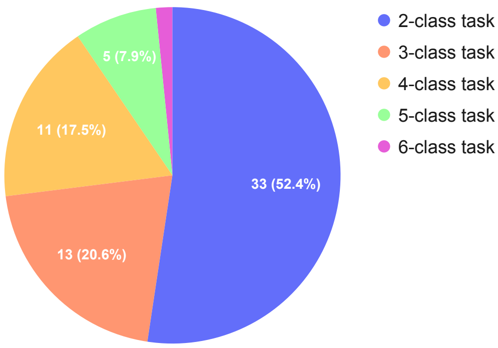


Figure 4: Distribution of 63 style classification tasks in §4.

C.4 Statistics and Examples of Generated Data

Figure 4 plots the distribution of 63 generated style classification tasks in this data generation process. We present examples of style annotation data and their lexicons in Table 15.

D Implementation Details

We use PyTorch (Paszke et al., 2019) and Huggingface Transformers (Wolf et al., 2020) in the experiments. In our zero-shot learning experiments, we prompted LLaMA-2-Chat (13B) to predict the target styles without any fine-tuning. We employed 4-bit inference due to our computing resource constraints (Dettmers and Zettlemoyer, 2023). In the zero-shot cross-style experiments, we first fine-tuned a model on the training styles before evaluating it on the evaluation styles. We fine-tuned the LLaMA-2 (7B) model on 4 A40 GPUs using DeepSpeed. All the other models were fine-tuned on one single A40 GPU. Hyperparameters are selected following the common practices in previous research. Table 16 reports the hyperparameters for our instruction tuning.

E Additional Experimental Results & Analyses

E.1 Impact of Instruction Templates

Prior works find that prompting an LLM on an unseen task is extremely sensitive to the prompt design, such as the wording of prompts (Sanh et al., 2021). To investigate the sensitivity of lexicon-based instructions, we experiment with four instruction templates `t1`, `t2`, `t3`, `t4` (Table 20), each of which contains different natural language task instructions. For each template, we fine-tune a model on our benchmark training styles using lexicon-based instructions. Table 17 shows that without randomization during instruction tuning, lexicon-based instruction (i.e., the “Lang” variant) is sensitive to the choice of templates. However, after introducing class randomization, lexicon-based instruction (i.e., the “Lang, Rw” variant) improves the average F1 across the templates by a substantial margin, while reducing the standard deviation, indicating that it is more robust to the wordings of the prompts.

Instruction Template in Main Experiments In our main experiments (§3), we conduct a comparative analysis between the lexicon-based instruction and the standard instruction. Both utilize the template **t2** in Table 20 except that the standard instruction does not incorporate any lexicon sampling. Instead, each slot for the lexicon words contains only the corresponding class name. Here is an example input of the standard instruction on Politeness: *In this task, you are given sentences. The task is to classify a sentence as "polite" if the style of the sentence is similar to the words "polite" or as "impolite" if the style of the sentence is similar to the words "impolite". Here is the sentence: "I've just noticed I wrote... and smooth out the text?"*. Its output is *polite*.

E.2 Impact of Lexicon Source

We study the impact of lexicon choices in lexicon-based instruction that include: (1) `dict`: all lexicons are from dictionary; (2) `nlp+chat`: for classes that have NLP lexicons, we directly use them, whereas for those without, we create ones using ChatGPT; (3) `class`: each class lexicon contains only its class name, e.g., the “humorous” lexicon has a single word “humorous”; (4) `human`: we have a native speaker create a lexicon for each style class, by carefully choosing words or phrases that best capture the characteristics of each style class. Table 17 shows that without class randomization during instruction tuning with lexicon, the average F1 for `nlp+chat` across four templates is the highest at 40.54. With randomization, `dict` performs the best at 54.50. Randomizing classes with words in lexicon-based instructions consistently improves the average F1 while reducing the standard deviation across four lexicon sources, regardless of the prompt templates used. The human-created lexicon is the most robust to the change of templates.

Style Classes and their Lexicons	Example	Label
helpful: supportive, wanting to help **unhelpful**: perfunctory, unfavorable	Okay, save it. I don't have time to hear your complaints.	unhelpful
	Person A: "I've been having a hard time getting over my ex." Person B: "Healing takes time, and it's okay to grieve a relationship. If you need someone to talk to, I'm here for you, anytime."	helpful
acrostic: nitials, word puzzle, creative **ghazal**: lyrical, emotive, spiritual **limerick**: humorous, rhythmic, short	There once was a man from Nantucket Who kept all his cash in a bucket. But his daughter, named Nan, Ran away with a man And as for the bucket, Nantucket.	limerick
	I am lost in love's reality, and I see you in dreams, In the silence of the night, in the roar of the streams, it's you.	ghazal
	Caring and kind, Always in my mind. Today and tomorrow, Heart full of sorrow. Yearning for your touch.	acrostic
supportive: empathetic, encouraging, comforting, helpful **unsupportive**: distant, dismissive, uncaring, brief	I believe in your abilities and I know you can do it.	supportive
	That's not up to the mark. You need to work harder.	unsupportive
philosophical: relating to the fundamental nature of knowledge, reality, and existence **inspirational**: providing creative or spiritual inspiration **funny**: humorous, causing laughter or amusement	It does not matter how slowly you go as long as you do not stop.	inspirational
	The unexamined life is not worth living.	philosophical
	I find television very educating. Every time somebody turns on the set, I go into the other room and read a book.	funny
condescending: patronizing, arrogant, superior **respectful**: polite, considerate, humble	Wow, you actually understood that concept? I'm impressed.	condescending
	Your social life seems vibrant and you're also doing well in your work. How do you manage that?	respectful

Table 15: Examples of new styles and instances generated semi-automatically using LLMs. These styles are used in §4 to further demonstrate the generalization ability of lexicon-based instructions.

Hyperparameter	$T5_{base}$	GPT-J	LLaMA-2 7B
optimizer	Adafactor	Adam	Adam
learning rate	1e-4	1e-5	2e-5
batch size	8	4	128
max encoder/input length	512	512	512
max decoder/target length	16	16	
# epochs	Instruction with class randomization: 5 Others: 3	1	3

Table 16: Hyperparameters of instruction tuning on the benchmark training styles. Note that the number of epochs depends on the model convergence rate. Instruction with class name randomization converge more slowly than the other prompts, so their epoch is longer.

E.3 Varying Number of Lexicon Words (m) in Lexicon-Based Instructions

When predicting a style in the evaluation split zero-shot, the lexicon instruction-tuned model only has access to a subset of m lexicon words that express or imply the style classes rather than example sentences. To investigate the model's dependence on the number of lexicon words, we take the variant of lexicon-based instruction with class randomization (i.e., the "Lang, Rw" variant) and incrementally increase m from 0 to 30 in both fine-tuning and evaluation phases. Figure 7 shows a general trend that the average F1 of six targets initially increases with increasing m, but then either drops or stabilizes. On average, our method performs the best when $m = 5$.

Moreover, we fix the model fine-tuned with the "Lang, Rw" lexicon-based instruction variant at $m = 5$, and then gradually increase m while evaluating evaluation styles. A similar trend is noticed in Figure 7. It can also be seen that when target styles have no lexicon resources ($m = 0$), increasing the number of lexicon words in each prompt during source fine-tuning might be beneficial. For instance, "src-5, tgt-0" improves the performance of "src-0, tgt-0" by an average of 3.96 F1 points.

Figure 8 provides a detailed view of the performance change associated with an increase in m, broken down by each target style. It reveals that different styles reach their peak performance at different values of m.

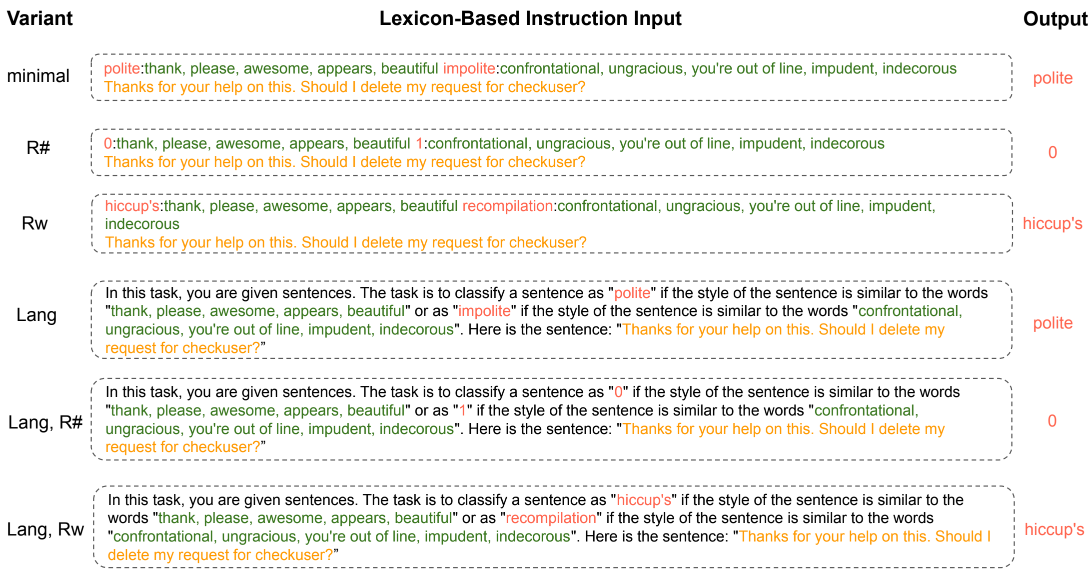


Figure 5: Examples of different lexicon-based instruction variants (as detailed in §2.3) on *Politeness*. Red part is (randomized) classes, the green part represents the words sampled from each class lexicon, and yellow stands for the input sentence and the uncolored part is the instruction template.

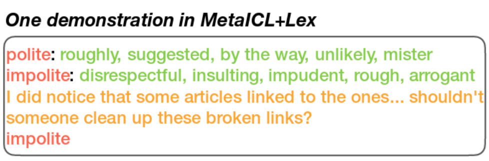


Figure 6: MetaICL+Lex input consists of K demonstrations and an input sentence. Each demonstration contains m lexicon words for each class, followed by an example with its label.

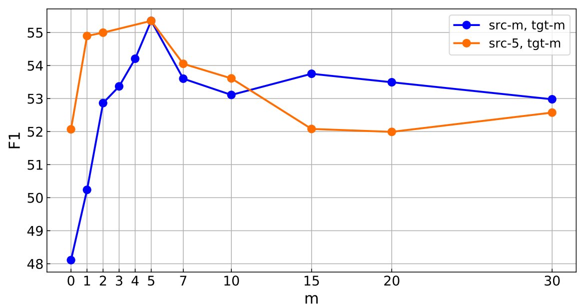


Figure 7: Impact of the number (m) of lexicon words or phrases used in each lexicon-based instruction. "src-m" is for fine-tuning on source styles (i.e., training styles) and "tgt-m" for evaluation on targets.

E.4 More Experiments on Style Splits

This section presents additional experimental results of our approach, utilizing various style splits outlined in Table 18. Results are presented in Table 19. It is observed that lexicon-based instruction tuning consistently outperforms standard instruction tuning across various style splits in both T5 and GPT-J models.

E.5 Comparisons of MetaICL and Lexicon-Based Instructions in Few-Shot Learning

To compare lexicon-based instructions and MetaICL fairly, it is necessary to incorporate supervision from K demonstrations in evaluation style into our approach. We thus introduce a modification to lexicon-based instructions called +Lex +K. Specifically, for each evaluation style, we randomly select K examples from its train set and assign a label to each. Next, a model that was previously fine-tuned on the training styles using the 'Lang, Rw' lexicon-based instructions, is further fine-tuned on these K demonstrations. Finally, we evaluate the model on the evaluation style using lexicon-based instructions without demonstrations.

The results are reported in Table 21. It is ob-

		dict	nlp+chat	class	human	**Avg.**	**SD.**
w/o rand. (*Lang*)	t1	42.55	43.88	41.99	41.64	42.52	0.99
	t2	39.05	41.72	33.71	41.56	39.01	3.74
	t3	35.40	40.21	36.13	38.69	37.61	2.24
	t4	30.43	36.33	37.02	36.14	34.98	3.06
	Avg.	36.86	40.54	37.21	39.51		
	SD.	5.18	3.18	3.48	2.63		
w/ rand. (*Lang, Rw*)	t1	54.20	54.72	53.16	55.15	54.31	0.86
	t2	54.74	54.23	50.24	54.83	53.51	2.20
	t3	53.24	52.17	51.59	53.85	52.71	1.02
	t4	55.83	51.89	55.02	53.91	54.16	1.71
	Avg.	54.50	53.25	52.50	54.44		
	SD.	1.08	1.43	2.06	0.66		

Table 17: For each combination of the lexicon source and the prompt template, class randomization (i.e., the "Lang, Rw" variant) consistently improves the average F1 scores. t1, t2, t3 and t4 are the different templates detailed in Table 20. dict, nlp+chat, class and human are the different lexicon sources described in Appendix E.2. Each white cell reports the result averaged over the six target styles. Light grey cells indicate the average (Avg.) and the standard deviation (SD.) scores over four lexicon sources. Dark grey cells represent Avg. and SD. over four templates.

Split	Source Styles
$\text{style}_{\text{src1}}$	Politeness, Formality, Sentiment
$\text{style}_{\text{src2}}$	Politeness, Formality, Sentiment, Hate/Offense
$\text{style}_{\text{src3}}$	Politeness, Formality, Sentiment, Hate/Offense, Politics
$\text{style}_{\text{src4}}$	Politeness, Formality, Sentiment, Hate/Offense, Politics, Readability, Subjectivity

Table 18: Source styles used in different source-target style splits.

served that with random labels, +Lex +K generally outperforms other methods. These may suggest that lexicons can provide a useful signal for the prediction of unseen styles when the gold labels of examples are absent.

E.6 Varying Number of Training Examples (K) used in Few-Shot Learning

We investigate the impact of the number of examples (K) that are used in the few-shot learning methods MetaICL_K and +Lex +K. Results are reported in Figure 9. The performance of both methods deteriorates with an increase in K when using random labels. However, when gold labels are utilized for the target-style training examples, the performance improves with larger K, particularly

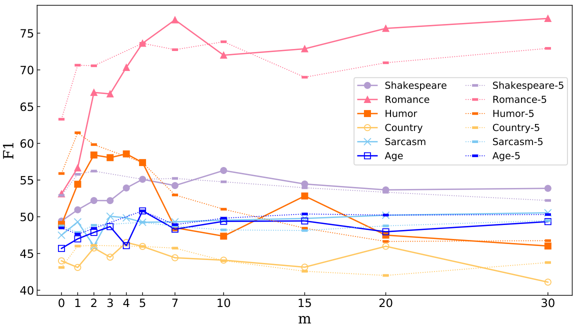


Figure 8: Impact of the number (m) of lexicon words or phrases used in each lexicon-based instruction. The solid lines represent the cases where m is applied to both source fine-tuning and target evaluation. The dotted lines (i.e., *Style*-5) show the scores of target styles when lexicon size 5 is used for source fine-tuning, while the size of target-style lexicons m is varied for evaluation.

Model	#Params	Instruction	$\text{style}_{\text{src1}}$	$\text{style}_{\text{src2}}$	$\text{style}_{\text{src3}}$	$\text{style}_{\text{src4}}$
T5	220M	Standard	36.72	36.27	30.01	33.72
		+ Lex	53.30	53.27	54.18	57.30
GPT-J	6B	Standard	50.14	53.64	56.06	51.96
		+ Lex	54.14	56.15	57.52	56.32

Table 19: Average F1 on the six evaluation styles. Across all training-evaluation splits, + Lex instruction improves the average performance on unseen styles compared to Standard instruction for both T5 and GPT-J.

showing significant improvement from $K = 8$ to $K = 16$. Moreover, as K increases, the performance disparity between utilizing ground-truth labels and random labels further expands. These observations show that the ground-truth input-label mapping is important in our case.

F More Prompting Examples

Figure 5 shows the example input and output for all lexicon-based instruction variants. In MetaICL_K+Lex, one prompt consists of K demonstrations and an input sentence. Figure 6 provides an example demonstration.

Instruction Template	Input	Output
t1	Which style best describes the sentence "{sentence}"? styles: - {className$_1$}: {$e_1, \cdots, e_k$} - {className$_2$}: {$e_1, \cdots, e_k$} ...	className$_i$
t2	In this task, you are given sentences. The task is to classify a sentence as "{className$_1$}" if the style of the sentence is similar to the words "{$e_1, \cdots, e_k$}" or as "{className$_2$}" if the style of the sentence is similar to the words "{$e_1, \cdots, e_k$}" or as $\cdots$ Here is the sentence: "{sentence}".	
t3	The task is to classify styles of sentences. We define the following styles: "{className$_1$}" is defined by "{$e_1, \cdots, e_k$}"; "{className$_2$}" is defined by "{$e_1, \cdots, e_k$}"; $\cdots$ Here is the sentence: "{sentence}", which is more like	
t4	Context: "{className$_1$}" is defined by "{$e_1, \cdots, e_k$}", "{className$_2$}" is defined by "{$e_1, \cdots, e_k$}" ... Sentence: {sentence} Question: which is the correct style of the sentence? Answer:	

Table 20: Instruction templates.

	Method	Shakespeare	Romance	Humor	Country	Sarcasm	Age	**Avg.**
Examples w/ *random* labels	MetaICL$_4$	44.37$_{\pm 6.99}$	56.21$_{\pm 26.64}$	37.82$_{\pm 5.02}$	41.84$_{\pm 18.46}$	35.55$_{\pm 2.94}$	40.96$_{\pm 11.19}$	42.79
	MetaICL$_4$+Lex	39.80$_{\pm 1.47}$	64.58$_{\pm 18.72}$	38.59$_{\pm 4.41}$	49.72$_{\pm 0.44}$	43.77$_{\pm 6.52}$	35.30$_{\pm 0.00}$	45.29
	+Lex +4	54.97$_{\pm 0.52}$	**83.63**$_{\pm 4.76}$	**58.11**$_{\pm 2.81}$	49.07$_{\pm 0.48}$	**47.98**$_{\pm 0.61}$	46.44$_{\pm 0.97}$	**56.70**
	MetaICL$_{16}$	55.49$_{\pm 11.66}$	66.91$_{\pm 20.48}$	36.11$_{\pm 4.58}$	7.74$_{\pm 4.67}$	33.33$_{\pm 0.00}$	31.24$_{\pm 0.00}$	38.47
	+Lex +16	**56.68**$_{\pm 2.71}$	66.87$_{\pm 17.72}$	57.69$_{\pm 1.93}$	**51.67**$_{\pm 0.76}$	45.67$_{\pm 3.71}$	**47.81**$_{\pm 1.62}$	54.40
Examples w/ *gold* labels	MetaICL$_4$	64.30$_{\pm 13.01}$	53.53$_{\pm 27.30}$	49.79$_{\pm 12.46}$	49.29$_{\pm 0.01}$	34.28$_{\pm 1.57}$	36.21$_{\pm 1.25}$	47.90
	MetaICL$_4$+Lex	43.90$_{\pm 8.06}$	75.80$_{\pm 6.52}$	42.78$_{\pm 3.99}$	49.42$_{\pm 0.36}$	38.62$_{\pm 3.69}$	35.30$_{\pm 0.00}$	47.63
	+Lex +4	54.42$_{\pm 1.78}$	85.48$_{\pm 3.00}$	58.83$_{\pm 4.93}$	48.92$_{\pm 0.43}$	43.11$_{\pm 4.91}$	45.84$_{\pm 1.94}$	56.10
	MetaICL$_{16}$	**72.93**$_{\pm 8.15}$	**95.79**$_{\pm 0.84}$	52.05$_{\pm 8.52}$	47.90$_{\pm 3.07}$	33.33$_{\pm 0.00}$	35.30$_{\pm 0.00}$	56.22
	+Lex +16	60.99$_{\pm 6.75}$	94.00$_{\pm 1.41}$	**63.26**$_{\pm 3.35}$	**51.85**$_{\pm 0.41}$	**44.93**$_{\pm 4.34}$	**47.42**$_{\pm 4.54}$	**60.41**

Table 21: Few-shot learning of GPT-J. The subscript of MetaICL represents the number (K) of demonstrations in one prompt. For each method (MetaICL$_K$, MetaICL$_K$+Lex, or +Lex +K), we choose a set of K examples with five different random seeds. By introducing lexicons into prompts, the standard deviation of performance across five runs generally decreases.

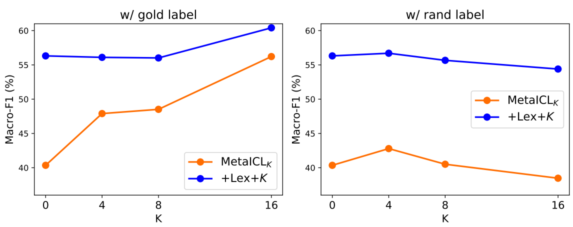


Figure 9: Ablation on the number of training examples (K) in a few-shot learning setting.