

Improving Formality Style Transfer with Context-Aware Rule Injection

Zonghai Yao

University of Massachusetts Amherst
zonghaiyao@cs.umass.edu

Hong Yu

University of Massachusetts Lowell
hong_yu@uml.edu

Abstract

Models pre-trained on large-scale regular text corpora often do not work well for user-generated data where the language styles differ significantly from the mainstream text. Here we present Context-Aware Rule Injection (CARI), an innovative method for formality style transfer (FST). CARI injects multiple rules into an end-to-end BERT-based encoder and decoder model. It learns to select optimal rules based on context. The intrinsic evaluation showed that CARI achieved the new highest performance on the FST benchmark dataset. Our extrinsic evaluation showed that CARI can greatly improve the regular pre-trained models' performance on several tweet sentiment analysis tasks.

1 Introduction

Many user-generated data deviate from standard language in vocabulary, grammar, and language style. For example, abbreviations, phonetic substitutions, Hashtags, acronyms, internet language, ellipsis, and spelling errors, etc are common in tweets (Ghani et al., 2019; Muller et al., 2019; Han et al., 2013; Liu et al., 2020). Such irregularity leads to a significant challenge in applying existing language models pre-trained on large-scale corpora dominated with regular vocabulary and grammar. One solution is using formality style transfer (FST) (Rao and Tetreault, 2018), which aims to transfer the input text's style from the informal domain to the formal domain. This may improve the downstream NLP applications such as information extraction, text classification and question answering.

A common challenge for FST is low resource (Wu et al., 2020; Malmi et al., 2020; Wang et al., 2020). Therefore, approaches that integrate external knowledge, such as rules, have been developed. However, existing work (Rao and Tetreault, 2018; Wang et al., 2019) deploy context-insensitive rule

injection methods (CIRI). As shown in Figure 1, when we try to use CIRI-based FST as the preprocessing for user-generated data in the sentiment classification task, according to the rule detection system, "extro" has two suggested changes "extra" or "extrovert" and "intro" corresponds to either "introduction" or "introvert." The existing CIRI-based FST models would arbitrarily choose rules following first come first served (FCFS). As such, the input "always, always they think I an extro, but Im a big intro actually" could be translated wrongly as "they always think I am an extra, but actually, I am a big introduction." This leads to the wrong sentiment classification since the FST result completely destroys the original input's semantic meaning.

In this work, we propose Context-Aware Rule Injection (CARI), an end-to-end BERT-based encoder and decoder model that is able to learn to select optimal rules based on context. As shown in Figure 1, CARI chooses rules based on context. With CARI-based FST, pre-trained models can perform better on the downstream natural language processing (NLP) tasks. In this case, CARI outputs the correctly translated text "they always think I am an extrovert, but actually, I am a big introvert," which helps the BERT-based classification model have the correct sentiment classification.

In this study, we performed both intrinsic and extrinsic evaluation of existing FST models and compared them with the CARI model. The intrinsic evaluation results showed that CARI improved the state-of-the-art results from 72.7 and 77.2 to 74.31 and 78.05, respectively, on two domains of a FST benchmark dataset. For the extrinsic evaluation, we introduced several tweet sentiment analysis tasks. Considering that tweet data is typical informal user-generated data, and regular pre-trained models are usually pre-trained on formal English corpora, using FST as a preprocessing step of tweet data is expected to improve the performance of reg-

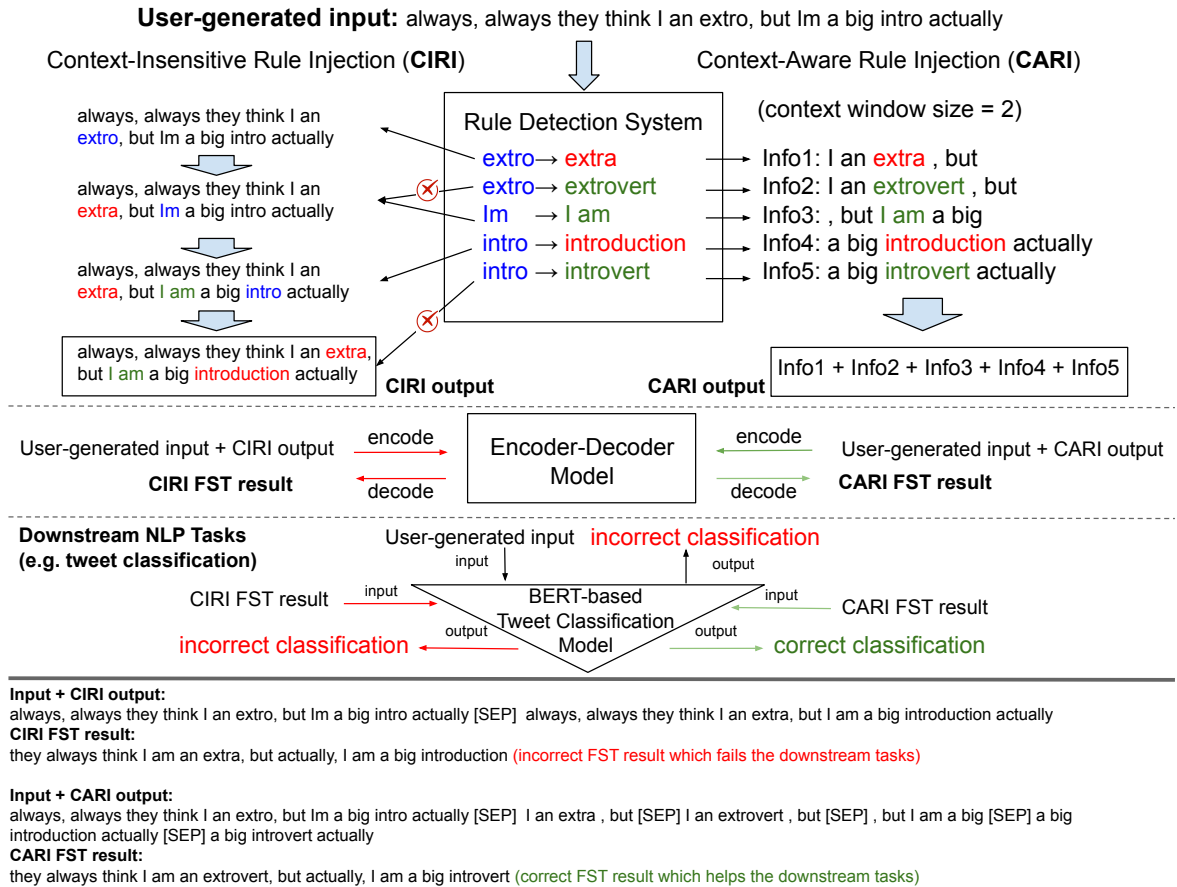


Figure 1: An example of using Context-Insensitive Rule Injection (CIRI) and Context-Aware Rule Injection (CARI) FST models. CIRI models are not context aware and therefore select rules arbitrarily and in this case, apply the rules First Come First Serve (FCFS). The errors introduced (“extra” and “introduction”) in the CIRI model impact the downstream NLP tasks, and in this case leading to the incorrect sentiment classification. In CARI, rules are associated with context and through training, CARI can learn to choose the right rules according to the context. This leads to improved FST thereby improves the downstream sentiment classification tasks.

ular pre-trained models on tweet downstream tasks. We regard measuring such improvement as the extrinsic evaluation. The extrinsic evaluation results showed that using CARI model as the preprocessing step improved the performance for both BERT and RoBERTa on several downstream tweet sentiment classification tasks. Our contributions are as follows:

1. We propose a new method, CARI, to integrate rules for pre-trained language models. CARI is context-aware and can be trained end-to-end with the downstream NLP applications.
2. We have achieved new state-of-the-art results for FST on the benchmark GYAFC dataset.
3. We are the first to evaluate FST methods with extrinsic evaluation and we show that CARI outperformed existing rule-based FST approaches for sentiment classification.

2 Related work

Rule-based Formality Style Transfer In the past few years, style-transfer generation has attracted increasing attention in NLP research. Early work transfers between modern English and the Shakespeare style with a phrase-based machine translation system (Xu et al., 2012). Recently, style transfer has been more recognized as a controllable text generation problem (Hu et al., 2017), where the style may be designated as sentiment (Fu et al., 2018), tense (Hu et al., 2017), or even general syntax (Bao et al., 2019; Chen et al., 2019). Formality style transfer has been mostly driven by the Grammarly’s Yahoo Answers Formality Corpus (GYAFC) (Rao and Tetreault, 2018). Since it is a parallel corpus, FST usually takes a seq2seq-like approach (Niu et al., 2018; Xu et al., 2019). Existing research attempts to integrate the rules into the

model because the GYAFC is low resource. However, rule matching and selection are context insensitive in previous methods (Wang et al., 2019). This paper focuses on developing methods for context-aware rule selection.

Evaluating Style Transfer Previous work on style transfer (Xu et al., 2012; Jhamtani et al., 2017; Niu et al., 2017; Sennrich et al., 2016a) has repurposed the machine translation metric BLEU (Papineni et al., 2002) and the paraphrase metric PINC (Chen and Dolan, 2011) for evaluation. Xu et al. (2012) introduced three evaluation metrics based on cosine similarity, language model and logistic regression. They also introduced human judgments for adequacy, fluency and style (Xu et al., 2012; Niu et al., 2017). Rao and Tetreault (2018) evaluated formality, fluency and meaning on the GYAFC dataset. Recent work on the GYAFC dataset (Wang et al., 2019; Zhang et al., 2020) mostly used BLEU as the evaluation metrics for FST. However, all aforementioned work focused on intrinsic evaluations. Our work has in addition evaluated FST extrinsically for downstream NLP applications.

Lexical Normalisation Lexical normalisation (Han and Baldwin, 2011; Baldwin et al., 2015) is the task of translating non-canonical words into canonical ones. Like FST, lexical normalisation can also be used to preprocess user-generated data. The MoNoise model (van der Goot and van Noord, 2017) is a state-of-the-art model based on feature-based Random Forest. The model ranks candidates provided by modules such as a spelling checker (aspell), a n-gram based language model and word embeddings trained on millions of tweets. Unlike FST, MoNoise and other lexical normalisation models can not change data’s language style. In this study, we explore the importance of language style transfer for user-generated data by comparing the results of MoNoise and FST models on tweets NLP downstream tasks.

Improving language models’ performance for user-generated data User-generated data often deviate from standard language. In addition to the formality style transfer, there are some other ways to solve this problem (Eisenstein, 2013). Fine-tuning on downstream tasks with a user-generated dataset is most straightforward, but this is not easy for many supervised tasks without a large amount of accurately labeled data. Another method is to fine-tune pre-trained models on the target domain

corpora (Gururangan et al., 2020). However, it also requires sizable training data, which could be resource expensive (Sohoni et al., 2019; Dai et al., 2019; Yao et al., 2020).

3 Approach

For the downstream NLP tasks where input is user-generated data, we first used the FST model for preprocessing, and then fine-tuned the pre-trained models (BERT and RoBERTa) with both the original data D_{ori} and the FST data D_{FST} , which were concatenated with a special token $[SEP]$, forming an input like $(D_{ori}[SEP]D_{FST})$.

For the formality style transfer task, we use the BERT-initialized encoder paired with the BERT-initialized decoder (Rothe et al., 2020) as the Seq2Seq model. All weights were initialized from a public BERT-Base checkpoint (Devlin et al., 2019). The only variable that was initialized randomly is the encoder-decoder attention. Here, we describe CARI and several baseline methods of injecting rules into the Seq2Seq model.

3.1 No Rule (NR)

First we fine-tuned the BERT model with only the original user-generated input. Given an informal input x_i and formal output y_i , we fine-tuned the model with $\{(x_i, y_i)\}_{i=0}^M$, where M is the number of data.

3.2 Context Insensitive Methods

For baseline models, we experimented with two state-of-the-art methods for injecting rules. We followed Rao and Tetreault (2018) to create a set of rules to convert original data x_i to preprocessed data x'_i by rules, and then fine-tune the model with parallel data $\{(x'_i, y_i)\}_{i=0}^M$. This is called **Rule Base (RB)** method. The preprocessed data, however, serves as a Markov blanket, i.e., the system is unaware of the original data, provided that only the preprocessed one is given. Therefore, the rule detection system could easily make mistakes and introduce noise.

Wang et al. (2019) improved the RB by concatenating the original text x_i with the text processed by rules x'_i with a special token $[SEP]$ in between, forming an input like $(x_i [SEP] x'_i)$. In this way, the model can make use of a rule detection system but also recognize its errors during the fine-tuning. This is called **Rule Concatenation (RCAT)** method. However, both RB and RCAT

methods are context insensitive, the rules were selected arbitrarily. In Figure 1 CIRI part, "extra" and "introduction" were incorrectly selected. This greatly limits the performance of the rule-based methods.

3.3 Context-Aware Rule Injection (CARI)

As shown in Figure 1, the input of CARI consists of the original sentence x_i and supplementary information. Suppose that r_i is an exhaustive list of the rules that are successfully matched on x_i . We make $r_i = \{(t_{i,j}, c_{i,j}, a_{i,j})\}_{j=0}^N$, where N is the total number of matched rules in r_i . Here, $t_{i,j}$ and $c_{i,j}$ are the corresponding matched text and context in the original sentence, respectively, for every matched rule in r_i , and $a_{i,j}$ are the corresponding alternative texts for every matched rule in r_i . Each supplementary information is composed of one alternative text $a_{i,j}$ and its corresponding context $c_{i,j}$. We connect all the supplementary information with the special token $[SEP]$ and then connect it after the original input. In this way, we form an input like $(x_i [SEP] a_{i,1}, c_{i,1} [SEP] \dots [SEP] a_{i,j}, c_{i,j})$. Finally, the concatenated sequence and the corresponding formal reference y_i serve as a parallel text pair to fine-tune the Seq2Seq model. Like RCAT, CARI can also use rule detection system and recognize its errors during the fine-tuning. Furthermore, since we keep all rules in the input, CARI is able to dynamically identify which rule to use, maximizing the use of the rule detection system.

4 Experimental setup

4.1 Datasets

For the intrinsic evaluation, we used the GYAFC dataset.¹ It consists of handcrafted informal-formal sentence pairs in two domains, namely, Entertainment & Music (E&M) and Family & Relationship (F&R). Table 1 shows the statistics of the training, validation, and test sets for the GYAFC dataset. In the validation and test sets of GYAFC, each sentence has four references. For better exploring the data requirements of different methods to combine rules, we followed Zhang et al. (2020) and used the back translation method (Sennrich et al., 2016b) to obtain additional 100,000 data for training. For rule detection system, we used the grammarbot API,² and Grammarly³ to help us create a set of rules.

¹<https://github.com/raosudha89/GYAFC-corpus>

²<https://www.grammarbot.io/>

³<https://www.grammarly.com/>

FST GYAFC dataset	Train	Valid	Test
Entertainment & Music	52,595	2,877	1,416
Family & Relationship	51,967	2,788	1,322

Affect in Tweets EI-oc	Train	Valid	Test
anger	1,701	388	1,002
fear	2,252	389	986
joy	1,616	290	1,105
sadness	1,533	397	975

Irony Detection	Train	Valid	Test
Irony-a	3067	767	784
Irony-b	3067	767	784

Table 1: The data statistics for GYAFC dataset of Formality style transfer task and Tweet NLP downstream classification datasets.

For the extrinsic evaluation, we used two datasets for sentiment classification: SemEval-2018 Task 1: Affect in Tweets EI-oc (Mohammad et al., 2018), and Task 3: Irony Detection in English Tweets (Van Hee et al., 2018). Table 1 shows the statistics of the training, validation, and test set for the two datasets. We normalized two tweet NLP classification datasets by translating word tokens of user mentions and web/url links into special tokens @USER and HTTPURL, respectively, and converting emotion icon tokens into corresponding strings.

4.2 Fine-tuning models

We employed the transformers library (Wolf et al., 2019) to independently fine-tune the BERT-based encoder and decoder model for each method in 20,000 steps (intrinsic evaluation), and fine-tune the BERT-based and RoBERTa-based classification models for each tweet sentiment analysis task in 10,000 steps (extrinsic evaluation). We used the Adam algorithm (Kingma and Ba, 2014) to train our model with a batch size 32. We set the learning rate to 1e-5 and stop training if validation loss increases in two successive epoch. We computed the task performance every 1,000 steps on the validation set. Finally, we selected the best model checkpoint to compute the performance score on the test set. We repeated this fine-tuning process three times with different random seeds and reported each final test result as an average over the test scores from the three runs. During inference, we use beam search with a beam size of 4 and beam

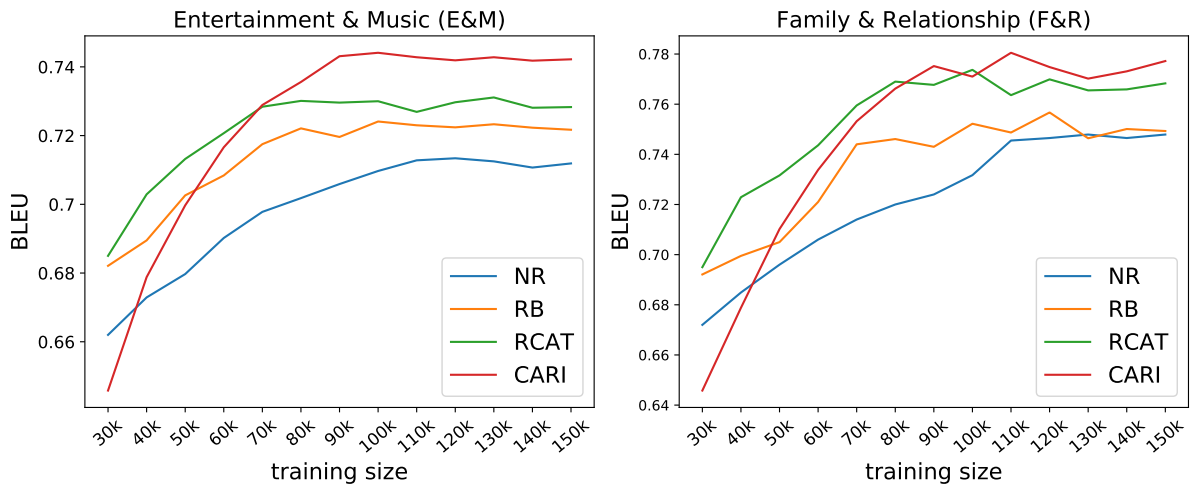


Figure 2: The performance (BLEU) of different rule injection methods with different training size. The results show that: 1) CARI achieved the best results in both E&M and F&R domains. 2) Rb, RCAT and CARI achieved optimal performance on less training size compared with NR, indicating the advantages of integrating rules to mitigate the low resource challenge. 3) compared with Rb and RCAT, CARI required slightly larger training size due to its context-aware learning model.

width of 6 to generate sentences. The whole experiment is carried out on 1 TITANX GPU. Each FST model finished training within 12 hours.

4.3 Intrinsic Evaluation Baselines

We used two state-of-the-art models, which were also relevant to our methods, as the strong intrinsic baseline models.

ruleGPT Like RCAT, Wang et al. (2019) aimed to solve the problem of information loss and noise caused by directly using rules as normalization in preprocessing. They put forward the GPT (Radford et al., 2019) based methods to concatenate the original input sentence and the sentence preprocessed by the rule detection system. Like the CIRI methods (RB, RCAT), their methods could not make full use of rules since they were also context-insensitive when selecting rules.

BT + M-Task + F-Dis Zhang et al. (2020) used three data augmentation methods, Back translation (Sennrich et al., 2016b), Formality discrimination, and Multi-task transfer to solve the low-resource problem. In our experiments, we also use the back translation method to obtain additional data because we want to verify the impact on the amount of training data required when using different methods to combine rules.

4.4 Extrinsic Evaluation Baselines

BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) are two typical regular language models pre-trained on large-scale regular formal text corpora, like BooksCorpus (Zhu et al., 2015) and English Wikipedia. The user-generated data, such as tweets, deviate from the formal text in vocabulary, grammar, and language style. As a result, regular language models often perform poorly on user-generated data. FST aims to generate a formal sentence given an informal one, while keeping its semantic meaning. A good FST result is expected to make regular language models perform better on user-generated data. For the extrinsic evaluation, we chose BERT and RoBERTa as the basic model. We introduced several tweet sentiment analysis tasks to explore the FST models' ability to transfer the user-generated data from the informal domain to the formal domain. Ideally, FST results for tweet data can improve the performance of BERT and RoBERTa on tweet sentiment analysis tasks. We regard measuring such improvement as the extrinsic evaluations. Besides, tweet data have much unique information, like Emoji, Hashtags, ellipsis, etc., which are not available in the GYAFC dataset. So in the extrinsic evaluation result analysis, although the final scores of FST-BERT and FST-RoBERTa were good, we paid more attention to the improvement of their performance before and after using FST, rather than the scores.

We used two different kinds of state-of-the-art

Irony Detection (evaluation metrics: F1)									
	UCDCC	BERT	MoNoise	RCAT	CARI	RoBERTa	MoNoise	RCAT	CARI
Irony-a	72.4	71.8	72.2	72.2	72.5	72.6	72.6	73.1	73.7
Irony-b	50.7	48.6	48.8	50.2	50.9	51.2	51	53.3	53.8

Affect in Tweets EI-oc (evaluation metrics: Pearson r)									
	SeerNet	BERT	MoNoise	RCAT	CARI	RoBERTa	MoNoise	RCAT	CARI
Joy	72	69.1	68.6	69.7	70.4	71.8	71.5	72.9	73.5
Anger	70.6	71.6	71.7	71.9	72	72	71.7	72.3	72.2
Sad	71.7	66.8	66.4	67.4	68.3	68.2	68	69.1	70.1
Fear	63.7	66.9	66.8	67.1	69.2	69.8	69.4	70.5	71.4

Table 2: The extrinsic evaluation results on tweet sentiment analysis tasks. Through observation, we can find that 1) Compared with the previous state-of-the-art results, the results of using BERT and RoBERTa directly were often very poor. 2) Monoise can not effectively improve the results of BERT and RoBERTa, while FST method can. 3) Compared with RCAT, CARI can better improve the results of BERT and RoBERTa on user-generated data.

Model	E&M	F&R
	BLEU	BLEU
no edit	50.28	51.67
ruleGPT	72.7	77.26
BT + M-Task + F-Dis	72.63	77.01
NR	71.94	75.65
RB	72.01	75.67
RCAT	73.01	77.37
CARI	74.31	78.05

Table 3: The comparison of our approaches to the state-of-the-art results on the GYAFC test set.

methods as our extrinsic evaluation baselines.

SeerNet and UCDCC We used the best results in the SemEval-2018 workshop as the first comparison method. For the task Affect in Tweets EI-o, the baseline is SeerNet (Duppada et al., 2018), and for the task Irony Detection in English Tweets, the baseline is UCDCC (Ghosh and Veale, 2018).

MoNoise MoNoise (van der Goot and van Noord, 2017) is the state-of-the-art model for the lexical normalization (Baldwin et al., 2015), which aimed to translate non-canonical words into canonical ones. Like the FST model, MoNoise can also be used as the preprocessing step in tweet classification tasks to normalize tweet input. So we used MoNoise as another comparison method.

5 Experimental results

5.1 Intrinsic Evaluation

Figure 2 showed the validation performance on both the E&M and the F&R domain. Compared to

the NR, the RB did not significantly improve. As we discussed above, even though the rule detection system will bring some useful information, it will also make mistakes and introduce noise. RB has no access to the original data, so it cannot distinguish helpful information from noise and mistakes. On the contrary, both RCAT and CARI have access to the original data, so their results improved a lot compared with RB. CARI had a better result compared to the RCAT. This is because RCAT is context insensitive while CARI is context-aware when selecting rules to modify the original input. Therefore, CARI is able to learn to select optimal rules based on context, while RCAT may miss using many correct rules with its pipeline preprocessing step for rules.

Figure 2 also showed the relationship between the different methods and the different training size. Compared with the NR method, the three methods which use rules can reach their best performance with smaller training size. This result showed the positive effect of adding rules in the low-resource situation of the GYAFC dataset. Moreover, CARI used larger training set to reach its best performance than RB and RCAT, since it needed more data to learn how to dynamically identify which rule to use.

In Table 4, we explored how large the context window size was appropriate for the CARI method on GYAFC dataset. The results showed that for both domains when the window size reaches two (taking two tokens each from the text before and after), Seq2Seq model can well match all rules with the corresponding position in the original input and

	context window size for CARI					
	0	1	2	3	4	5
E&M	68.1	72.5	74.2	74.6	74.3	74.5
F&R	70.5	74.3	76.9	77.5	76.8	77.3

Table 4: CARI performance (BLEU) by different context window size. When the context window size reach 2, the model can make good use of the rules’ information.

select the correct one to use.

5.2 Extrinsic Evaluation

Table 2 showed the effectiveness of using the CARI as the preprocessing step for user-generated data on applying regular pre-trained models (BERT and RoBERTa) on the downstream NLP tasks.

Compared with the previous state-of-the-art results (UCDCC and SeerNet), the results of using BERT and RoBERTa directly were often very poor, since BERT and RoBERTa were only pre-trained on regular text corpora. Tweet data has the very different vocabulary, grammar, and language style from the regular text corpora, so it is hard for BERT and RoBERTa to have good performance with small amount of fine-tuning data.

The results of RCAT and CARI showed that FST can help BERT and RoBERTa improve their performance on tweet data, because they can transfer tweets into more formal text while keeping the original intention as much as possible. CARI performed better than RCAT, which was also in line with the results of intrinsic evaluation. This result also showed the rationality of our extrinsic evaluation metrics.

Comparing the results of MoNoise with BERT and RoBERTa, the input preprocessed by MoNoise can not help the pre-trained model to improve effectively. We think that this is because the lexical normalization models represented by MoNoise only translate non-canonical words on tweet data into canonical ones. Therefore, MoNoise can basically solve the problem of different vocabulary between regular text corpora and user-generated data, but it can not effectively solve the problem of different grammar and language style. As a result, for BERT and RoBERTa, even though there is no Out-of-Vocabulary (OOV) problem in the input data processed by MoNoise, they still can not accurately understand the meaning of the input.

This result confirmed the previous view that lexical normalization on tweets is a lossy trans-

lation task (Owoputi et al., 2013; Nguyen et al., 2020). On the contrary, the positive results of the FST methods also showed that FST is more suitable as the downstream task preprocessing step of user-generated data. Because FST models need to transfer the informal language style to a formal one while keeping its semantic meaning, which makes a good FST model can ideally handle all the problems from vocabulary, grammar, and language style. This can help most language models pre-trained on the regular corpus, like BERT and RoBERTa, perform better on user-generated data.

5.3 Manual Analysis

The prior evaluation results reveal the relative performance differences between approaches. Here, we identify trends per and between approaches. We sample 50 informal sentences total from the datasets and then analyze the outputs from each model. We present several representative results in Table 5.

Examples 1 and 2 showed that, for BERT and RoBERTa, FST models are more suitable for preprocessing user-generated data than lexical normalization models. In example 1, both methods can effectively deal with the problem at the vocabulary level (“2” to “to,” “ur” to “your,” and “U” to “you”). However, in example 2, FST can further transform source data into a more familiar language style for BERT and RoBERTa, which is not available in the current lexical normalization methods such as MoNoise.

Example 3 showed the importance of injecting rules into the FST models. The word “idiotic” is a misspelling of “idiotic,” which is an OOV. Therefore, without the help of rules, the model can not understand the source data’s meanings and produced the wrong final output “I do not understand your question.”

Example 4 showed the importance of context for rule selection. The word “concern” provides the required context to understand that “exo” refers to an “extra” ticket. So the CARI-based model can choose the right one (“exo” to “extra”).

Examples 5 and 6 showed the shortcomings of CARI. In example 5, the rule detection system did not provide the information that the “fidy center” should be “50 Cent (American rapper)”, so CARI delivered the wrong result. Even though CARI helps mitigate the data low resource challenge, it faces the challenge on its own. CARI depends

Example 1:	Source:	explain 2 ur parents that u really want 2 act !!!
	MoNoise:	explain to your parents that you really want to act !
	FST:	explain to your parents that you want to act .
Example 2:	Source:	my observation skills??? wow, very dumb.....
	MoNoise:	my observation skills ? wow, very dumb . very
	FST:	my observation skills are very bad .
Example 3:	Source:	hell no your idiotic for asking .
	NR:	i do not understand your question .
	CARI:	absolutely not and i feel you are idiotic for asking .
Example 4:	Source:	got exo to share, concert in hk ! u interested ?
	RCAT:	have you got exo to share, concert in hong kong . are you interested ?
	CARI:	i got extra to share , concert in hong kong . are you interested ?
Example 5:	Source:	fidy cent he is fine and musclar
	Target:	50 Cent is fine and muscular .
	CARI:	fidy cent is fine and muscular .
Example 6:	Source:	if my pet bird gets too flappy, my pet kitty cat might eaty
	Target:	if my pet bird gets too flappy, my pet kitty cat might eat it
	CARI:	if my pet bird gets too flappy, my pet kitty cat might eat me

Table 5: Sample model outputs. Example 1 shows that both MoNoise and FST models can handle some simplest modifications. Example 2 shows that FST can transform the language style of user-generated data, while MoNoise can not. Example 3 shows that FST can not understand the source because of OOV noises in the data, while CARI-based FST can understand with rules. Example 4 shows the importance of context for rule selection. The word "concern" provides the required context to understand that "exo" refers to an "extra" ticket. In example 5, the rule detection system did not provide the information that the "fidy center" should be "50 Cent (American rapper)", so CARI makes the wrong result. In example 6, CARI mistakenly selected the rule "eat me."

on the quality of the rules, and in this case, no rule exists that links "fidy" to "50." In example 6, CARI mistakenly selected the rule "eat me," but not "eat it." This example also demonstrates the data sparsity that CARI faces. Here "eat me" is more commonly used than "eat it."

6 Conclusions

In this work, we proposed the Context-Aware Rule Injection(CARI), an innovative method for formality style transfer (FST) by injecting multiple rules into an end-to-end BERT-based encoder and decoder model. The intrinsic evaluation showed our CARI method achieved the highest performance with previous metrics on the FST benchmark dataset. Besides, we were the first to evaluate FST methods with extrinsic evaluation and specifically on the sentiment classification tasks. The extrinsic evaluation results showed that using the CARI-based FST as the preprocessing step outperformed existing rule-based FST approaches. Our results showed the rationality of adding such extensive evaluation.

Acknowledgments

The authors are grateful to Hadi Amiri (University of Massachusetts, Lowell) for his expert help in processing Twitter data, and to UMass BioNLP Group for lots of meaningful discussions.

This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

- Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135.
- Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xinyu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. *arXiv preprint arXiv:1907.05789*.

- David Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200.
- Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. *arXiv preprint arXiv:1906.00565*.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Venkatesh Duppada, Royal Jain, and Sushant Hiray. 2018. Seernet at semeval-2018 task 1: Domain adaptation for affect in tweets. *arXiv preprint arXiv:1804.06137*.
- Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 359–369.
- Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Norjihan Abdul Ghani, Suraya Hamid, Ibrahim Abaker Targio Hashem, and Ejaz Ahmed. 2019. Social media big data analytics: A survey. *Computers in Human Behavior*, 101:417–428.
- Aniruddha Ghosh and Tony Veale. 2018. Ironymagnet at semeval-2018 task 3: A siamese network for irony detection in social media. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 570–575.
- Rob van der Goot and Gertjan van Noord. 2017. Monoise: Modeling noise using a modular normalization system. *arXiv preprint arXiv:1710.03476*.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don’t stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.
- Bo Han and Timothy Baldwin. 2011. Lexical normalization of short text messages: Makn sens a# twitter. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 368–378.
- Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):1–27.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596. PMLR.
- Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv preprint arXiv:1707.01161*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Yinan Liu, Wei Shen, Zonghai Yao, Jianyong Wang, Zhenglu Yang, and Xiaojie Yuan. 2020. Named entity location prediction combining twitter and web. *IEEE Transactions on Knowledge and Data Engineering*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. *arXiv preprint arXiv:2010.01054*.
- Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.
- Benjamin Muller, Benoît Sagot, and Djamel Seddah. 2019. Enhancing bert for lexical normalization. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 297–306.
- Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*.
- Xing Niu, Marianna Martindale, and Marine Carpuat. 2017. A study of style in machine translation: Controlling the formality of machine translation output. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2814–2819.

- Xing Niu, Sudha Rao, and Marine Carpuat. 2018. Multi-task neural models for translating between styles within and across languages. *arXiv preprint arXiv:1806.04357*.
- Olutobi Owoputi, Brendan O’Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 380–390.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *Technical report, OpenAI*.
- 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.
- Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Nimit Sharad Sohoni, Christopher Richard Aberger, Megan Leszczynski, Jian Zhang, and Christopher Ré. 2019. Low-memory neural network training: A technical report. *arXiv preprint arXiv:1904.10631*.
- Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50.
- Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhao Chao. 2019. [Harnessing pre-trained neural networks with rules for formality style transfer](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3573–3578, Hong Kong, China. Association for Computational Linguistics.
- Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhao Chao. 2020. Formality style transfer with shared latent space. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2236–2249.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.
- Yu Wu, Yunli Wang, and Shujie Liu. 2020. A dataset for low-resource stylized sequence-to-sequence generation. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.
- Ruochen Xu, Tao Ge, and Furu Wei. 2019. Formality style transfer with hybrid textual annotations. *arXiv preprint arXiv:1903.06353*.
- Wei Xu, Alan Ritter, William B Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914.
- Zonghai Yao, Liangliang Cao, and Huapu Pan. 2020. Zero-shot entity linking with efficient long range sequence modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2517–2522.
- Yi Zhang, Tao Ge, and Xu Sun. 2020. Parallel data augmentation for formality style transfer. *arXiv preprint arXiv:2005.07522*.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.