Don't lose the message while paraphrasing: A study on content preserving style transfer

Nikolay Babakov[1⋆], David Dale[3,⋆], Ilya Gusev[3], Irina Krotova[4], and Alexander Panchenko[2,5]

[1] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela
[2] Skolkovo Institute of Science and Technology (Skoltech)
[3] Independent researcher
[4] Mobile TeleSystems (MTS)
[5] Artificial Intelligence Research Institute (AIRI)
a.panchenko@skol.tech

Abstract. Text style transfer techniques are gaining popularity in natural language processing allowing paraphrasing text in the required form: from toxic to neural, from formal to informal, from old to the modern English language, etc. Solving the task is not sufficient to generate *some* neural/informal/modern text, but it is important to preserve the original content unchanged. This requirement becomes even more critical in some applications such as style transfer of goal-oriented dialogues where the factual information shall be kept to preserve the original message, e.g. ordering a certain type of pizza to a certain address at a certain time. The aspect of content preservation is critical for real-world applications of style transfer studies, but it has received little attention. To bridge this gap we perform a comparison of various style transfer models on the example of the formality transfer domain. To perform a study of the content preservation abilities of various style transfer methods we create a parallel dataset of formal vs. informal task-oriented dialogues. The key difference between our dataset and the existing ones like GYAFC [17] is the presence of goal-oriented dialogues with predefined semantic slots essential to be kept during paraphrasing, e.g. named entities. This additional annotation allowed us to conduct a precise comparative study of several state-of-the-art techniques for style transfer. Another result of our study is a modification of the unsupervised method LEWIS [19] which yields a substantial improvement over the original method and all evaluated baselines on the proposed task.

Keywords: text style transfer · formality transfer · content preservation

1 Introduction

Text style transfer (**TST**) systems are designed to change the style of the original text to an alternative one, such as more positive [12], more informal [17], or

⋆ Work mostly has been done while at Skoltech even more Shakespearean [8]. Such systems are becoming very popular in the NLP. They could be applied to many purposes: from assistance in writing to diversifying responses of dialogue agents and creating artificial personalities.

Task-oriented dialogue agents are one of the possible applications of TST. In such dialogues, it is crucial to preserve important information such as product names, addresses, time, etc. Consider the task of making the source sentence from dialogue agent *Do you want to order a pizza to your office at 1760 Polk Street?* more informal to improve the user experience with the agent. This text contains named entities (*pizza*, *1760 Polk Street*) that are critical to understanding the meaning of the query and following the correct scenario and that could be easily lost or corrupted during standard beam search-based generation even if the model is trained on parallel data [1]. At the same time, there are several words in this sentence that could be changed to make the style more informal. For example, a target sentence such as *do u wanna order a pizza 2 ur office at 1760 Polk Street?* requires only small edition of some words not related to the important entities. This suggests that it could be better to keep the important entities intact and train the model to fill the gaps between them.

In this work, we focus on text formality transfer, or, more precisely, transferring text style from formal to informal with an additional requirement to preserve the predefined important slots. We assume that the transfer task is supervised, which means that a parallel corpus of the text pairs in the source and target style is available (we use the GYAFC dataset [17]).

A similar intuition has been used in the unsupervised TST domain in LEWIS [19], where the authors created a pseudo-parallel corpus, trained a RoBERTa [30] tagger to identify coarse-grain Levenshtein edit types for each token of the original text, and finally used a BART [10] masked language model to infill the final edits. With the increasing interest in the TST field, several large parallel datasets have been collected for the most popular TST directions, such as formality transfer [17]. Thus, it became possible to use the advantage of parallel data to address the specific task of content preservation.

The contributions of our work are three-fold:

1. We present *PreserveDialogue*: the first dataset for evaluating the content-preserving formality transfer model in the task-oriented dialogue domain.
2. We perform a study of strong supervised style transfer methods, based on transformer models, such as GPT2 and T5 (as well as simpler baselines), showing that methods based on Levenstein edit distance such as LEWIS [10] are outperforming them if content shall be strictly preserved.
3. We introduce LEWIT, an improved version of the original LEWIS model based on T5 encoder-decoder trained on parallel data which yields the best results across all tested methods.

We open-source the resulting dataset and the experimental code[6]. Additionally, we release the best-performing pre-trained model LEWIT for content preserving formality transfer on HugingFace model hub.[7]

[6] https://github.com/s-nlp/lewit-informal
[7] https://huggingface.co/s-nlp/lewit-informal

2 Related works

In this section, we briefly introduce the existing approaches to text generation with an emphasis on preserving certain content.

Constrained beam search The standard approach to text generation is beam search which iteratively generates possible next tokens, and the sequence yielding the highest conditional probability is selected as the best candidate after each iteration. There are several methods to constraint the beam search process which can be roughly divided into two broad categories: hard and soft constraints. In the hard constrained category, all constraints are ensured to appear in the output sentence, which is generally achieved by the modified type of beam search, allowing to directly specify the tokens to be preserved [7]. Opposite to hard-constrained approaches, soft-constrained approaches modify the model's training process by using the constraints as an auxiliary signal. Such signal is often either marked with special tags [11] or simply replaced with delexicalized tokens [5] during the training process and inference.

Edit based generation Beam search is not the only existing approach to text generation. One popular substitution is Levenstein transformer [6] — a partially autoregressive encoder-decoder framework based on Transformer architecture [23] devised for more flexible and amenable sequence generation. Its decoder models a Markov Decision Process (MDP) that iteratively refines the generated tokens by alternating between the insertion and deletion operations via three classifiers that run sequentially: deletion, a placeholder (predicting the number of tokens to be inserted), and a token classifier.

Content preservation in text style transfer Content preservation in text style transfer has mostly been addressed in the unsupervised domain. These methods mostly rely on text-editing performed in two steps: using one model to identify the tokens to delete and another model to infill the deleted text slots [26]. LEWIS [19] approach first constructs pseudo parallel corpus using an attention-based detector of style words and two style-specific BART [10] models, then trains a RoBERTa-tagger [30] to label the tokens (insert, replace, delete, keep), and finally fine-tunes style-specific BART masked language models to fill in the slots in the target style. LEWIT extrapolates this idea to a supervised setting. The main features of this work are that the token tagger is trained on tags obtained from parallel data, and the slots are filled with a T5 model [16], by taking advantage of its initial training task of slot filling.

3 Datasets

In this section, we describe the parallel training dataset and the evaluation dataset used respectively for tuning and evaluating the content-preserving formality transfer methods.

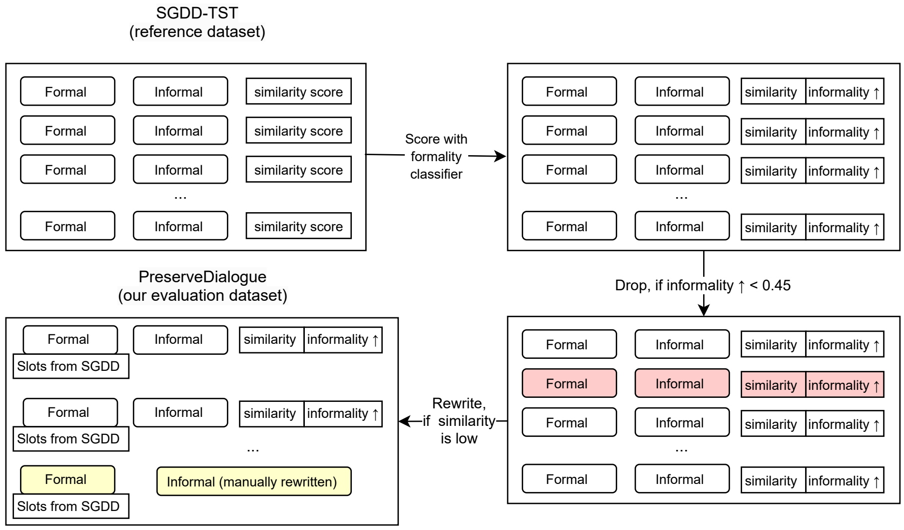

Fig. 1. The pipeline of collecting the PreserveDialogue dataset. The reference SGDD-TST dataset consists of formal-informal sentence pairs annotated with semantic similarity scores. The pairs are scored with the formality classifier model and the pairs with insignificant informality increase are dropped. The pairs with significant formality increase that have low semantic similarity scores are manually rewritten to be semantically similar. Finally, the important slots related to the formal sentence are extracted from the SGDD dataset.

3.1 Parallel training dataset: GYAFC

In terms of our work, we assume the availability of parallel data. We also focus our experiments on the transfer of formal text to a more informal form. Grammarly's Yahoo Answers Formality Corpus (GYAFC) containing over 110K informal/formal sentence pairs fits well to this task. The main topics of the sentences in this dataset are related to either entertainment and music or family and relationships, both of these topics take almost equal part in the dataset [17].

3.2 Parallel evaluation dataset: PreserveDialogue

We create a special evaluation dataset denoted as *PreserveDialogue*. It is based on SGDD-TST [1][8]. SGDD-TST consists of sentence pairs in a formal and informal style with human annotation of semantic similarity. Its formal phrases were obtained from SGDD [18] and informal ones were generated by a large T5-based model tuned on the GYAFC dataset. Some of the generated paraphrases were annotated as semantically different, which is why SGDD-TST in its original form is not appropriate for evaluating content-preserving style transfer. Thus we

[8] https://github.com/s-nlp/SGDD-TST

Source formal text	Important slots	Target informal rewrite
Red Joan sounds great	Red Joan	red joan is cool
What do I have scheduled Tuesday next week?	Tuesday next week	what stuff do i have to do on tuesday next week?
I am looking for a unisex salon in SFO.	SFO	i wanna find a unisex salon in SFO
Please confirm this: play Are You Ready on TV	Are You Ready	plz confirm this: play Are You Ready on TV
Where do you want to pick it up at?	–	where do u wanna pick it up at?

Table 1. Examples of texts from the PreserveDialogue dataset used for evaluation.

create PreserveDialogue as a derivative from SGDD-TST. Fig. 1 shows the main steps of the PreserveDialogue collection process. These steps are also described in more details below:

1. **Selecting sentences with significant informality increase.** We score all sentence pairs of SGDD-TST with a formality classifier (described in Section 4) and leave only 1100 pairs with an informality increase (namely, the difference between formality classifier scores of formal and informal sentences) greater than the empirically selected threshold of 0.45.
2. **Rewriting paraphrases with the corrupted sense.** Within the 1100 pairs, 369 are not semantically equal to each other (according to the similarity score available in the reference SGDD-TST dataset). Such pairs are rewritten by the members of our team according to the common intuition of informal style. After the messages are rewritten their informality is verified with the similar formality classifier used in the previous step.
3. **Extracting important slots.** The SGDD dataset [18] (the source of formal phrases of SGDD-TST) contains task-oriented dialogues with predefined named entities. We use them as important slots for the sentences in a formal style in PreserveDialogue.

Finally, the PreserveDialogue dataset consists of 1100 sentence pairs of formal and informal phrases. By the steps described above we make sure that these pairs have the equivalent sense and significantly differ in terms of informality. Moreover, each pair has a set of entity slots extracted from SGDD [18] (predecessor of SGDD-TST). These slots are related to the first sentence in the pair and are considered significant information which should be kept during formality transfer. A sample from the dataset can be found in Table 1.

4 Evaluation

In this section, we describe the methods of automatic evaluation of the formality transfer models.

In most TST papers (e.g. [28, 13, 22]) the methods are evaluated with the combination of the measures that score the basic TST properties: style accuracy, content preservation, and fluency. Our work is dedicated to formality transfer with an additional task to preserve particular slots. That is why we have to use an additional evaluation method: slot preservation. All of these measures evaluate TST quality from significantly different points of view, thus to reveal the TST method that performs best we aggregate them by *multiplying the four measures for each sentence and then averaging the products over the dataset*, following the logic of [9]. More details about each measure are provided in the following paragraphs. The code of measures calculations is also open-sourced in our repository.

Content preservation To score the general similarity between the generated text and the reference informal text we use Mutual Implication Score[9], a measure of content preservation based on predictions of NLI models in two directions. This measure has been compared [2] to a large number of SOTA content similarity measures and it was shown that it demonstrates one of the highest correlations with human judgments in the formality transfer domain: Spearman correlation between 0.62 and 0.77 depending on the dataset.

Slots preservation The key point of content preservation, especially in task-oriented dialogues, is keeping the important entities from a source sentence (see Section 3.2). Thus, we check whether these entities exist in the generated sentence. Most entities could have at least two different forms, which could be considered correct (e.g. "fourteen" and "14"). To ensure that the entity is not considered lost even if it is generated in an alternative form, we normalize the important slots and generated text in the following way. All text tokens are lowercased and lemmatized. The state names (e.g. "Los Angeles"—"LA"), numbers (e.g. "six"—"6"), and time values (e.g. "9am"—"9a.m."—"nine in the morning") are adjusted to a standard form using a set of rules. Some frequent abbreviations (geographic entity types, currencies) are expanded. The slots that still were not matched exactly are matched to the n-gram of the new sentence with the highest ChrF score [15]. The final slots preservation score is calculated as the ratio of the preserved slots in a new sentence (with ChrF scores as weights for the slots that were matched approximately) to the total number of slots in a source sentence. This ratio calculation takes into account both original and standardized forms of the tokens. This approach uses the idea similar to copy success rate calculation used for scoring constraints preservation in machine translation [4].

Style accuracy To ensure that the generated text corresponds to the target style we use a RoBERTA-based formality ranker[10]. The ranker was trained on two formality datasets: GYAFC [17] and P&T [14]. We verified the quality of

[9] https://huggingface.co/s-nlp/Mutual-Implication-Score
[10] https://huggingface.co/s-nlp/roberta-base-formality-ranker

this ranker by calculating the Spearman correlation of its score on the test split of GYAFC and P&T, which was 0.82 and 0.76 correspondingly.

Fluency The generated text should look natural, grammatical and fluent. Fluency is often evaluated as the perplexity of a large language model, but to make the results more interpretable, we use a RoBERTA-based classifier trained on the Corpus of Linguistic Acceptability (CoLA) [24]. It is a diverse dataset of English sentences annotated in a binary way, as grammatically acceptable or not. A detailed justification of using a CoLA-based classifier for fluency evaluation is presented in [9]. We use an opensource RoBERTA-based classifier[11] trained on CoLA for 5 epochs with a batch size of 32, a learning rate of 2e-05, and a maximum sequence length of 128. Its scores range from 0 to 1 with greater values meaning higher quality, just like all the other metrics we use for evaluation. The reported accuracy of this model on the CoLA validation set is 0.85.

5 Supervised Style Transfer Methods

In this section, we describe the baseline methods used in our computational study dedicated to finding the best approach to content-preserving formality transfer. All models requiring tuning described in this section are tuned on the GYAFC parallel dataset (see Section 3.1).

5.1 Naive baselines

We use two naive baselines. In *copy-paste*, we simply copying the source text, and in *only-slots*, the target string is simply a list of important slots separated by commas. The motivation of these methods is the sanity check of the proposed evaluation pipeline (see Section 4). The joint score (multiplication of four measures) is supposed to place the naive methods at the bottom of the leaderboard, which could be treated as a necessary condition of acceptance of the proposed evaluation method.

5.2 Sequence-to-sequence approaches

As the setting of our work assumes the availability of parallel data, it is natural to try standard sequence-to-sequence models (seq2seq), both “as is” and with some modifications related to the task of content and slots preservation.

Standard seq2seq We tune the following models in the standard seq2seq approach: pure T5-base (*seq2seq-t5*) and T5-base pre-tuned on a paraphrasing datasets[12] (*seq2seq-t5-para*). We also experiment with using a template generated from the target sentence as a text pair for the training of the T5 model (*seq2seq-t5-template*).

[11] https://huggingface.co/textattack/roberta-base-CoLA
[12] https://huggingface.co/ceshine/t5-paraphrase-paws-msrp-opinosis

seq2seq with hard lexical constraints The models trained in the standard seq2seq approach can be inferenced with lexically constrained beam search (*seq2seq-t5-para-constr*, *seq2seq-t5-constr*) which is implemented in the HuggingFace library mainly based on dynamic beam allocation[13] [7].

Re-ranking beam search outputs with a neural textual similarity metric. We experiment with re-ranking beam search outputs with neural textual similarity metric. The hypothesis obtained after the beam search could be reranked w.r.t. some content preservation measure. To avoid overfitting, we should not rerank with the same measure (MIS) that we use for evaluation. In [2], the authors show that apart from MIS, BLEURT [21] also demonstrates reasonable performance in the formality transfer domain. We use a mean of BLEURT-score and conditional probability to perform a final re-ranking of the hypothesis generated after the beam search. This approach is used in combination with seq2seq-para-constr (*rerank-BLEURT-constr*) and with seq2seq-para (*rerank-BLEURT*).

Learning to preserve slots with tags Finally, we try to embed the task of content preservation into the seq2seq training. One of the possible ways to do that is to embed a signal in the training data indicating that a certain slot should be preserved. We use two different types of such signals. First, similarly to the idea presented in [27] we put special `<tag>` tokens around the slots to be preserved (*slot-tags*). Second, we replace the whole slot with a placeholder token and train model to re-generate this placeholder, which is then filled with the value from the original sentence [5] (*delex*).

5.3 Language models inference

There exists some evidence of the possibility to use the large pre-trained language models (LM) in zero- and few-shot way [3]. The LM can also be slightly fine-tuned on the parallel data to be capable of performing the desired task.

Similarly to the idea of [20], we construct a prompt for the language model to make it generate more informal text: *"Here is a text, which is formal: <formal text>. Here is a rewrite of the text which contains <slot 1>, <slot 2> and is more informal"* and train GPT2-medium[14] on parallel data to continue this prompt. We use two variations of such approach: with (*GPT2-tuned-constr*) and without (*GPT2-tuned*) the information about constraints in the prompt.

5.4 LEWIS and its modifications

An intuitively straightforward approach for a human to generate an informal paraphrase is to apply some slight modifications to the formal source text. This

[13] https://github.com/huggingface/transformers/issues/14081
[14] https://huggingface.co/gpt2-medium

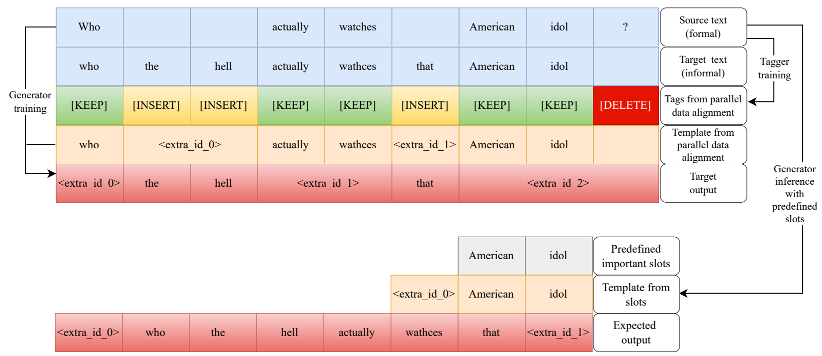

Fig. 2. LEWIT model workflow: The edit tags are obtained from the alignment of source formal and target informal texts. These tags are used to train the token tagger. These edit tags are also used to create a template used to train a T5-generator model, which fills the slots between the preserved tokens.

group of approaches is named "edit-based". Most of these approaches use numerous models to perform separate edition actions for generating a new text: deletion, insertion, placeholder, infiller models [29]. We experimented with the LEWIS model [19] representing this kind of methods.

LEWIS was designed in the unsupervised domain, so the authors first created a pseudo-parallel corpus, then trained a RoBERTa tagger to identify coarse-grain Levenshtein edit types for each token from the original text, and finally used a BART masked language model to infill the final edits. We use LEWIS in a parallel data setting by tuning BART on our parallel dataset and using it either with known constraints (*LEWIS-constr*) or with the labels inferred from the RoBERTa tagger trained on the edits from parallel data (*LEWIS-tag*).

We also test a modified version of LEWIS denoted as LEWIT (T5-based LEWIS): similarly to LEWIS architecture it involves a token tagger trained on Levenshtein edits obtained from the alignment of the parallel data, but its infiller model (i.e. the model inserting the tokens between other tokens) is based on T5 that was originally trained with the specific task of infilling gaps of several tokens. The LEWIT model consists of two steps as illustrated in Figure 2. First, the RoBERTa token tagger is trained on the tags from the GYAFC dataset (see Section 3.1) which are directly computed from edits required to transform the source texts into the target texts. Second, the T5-based[15] generator model is trained on the templates from parallel data that was also generated from the GYAFC dataset sentence pairs. The model receives the concatenation of the source sentence and the template constructed w.r.t. the edit tokens and is expected to generate the words masked from the target sentence.

LEWIT inherits the general logic of LEWIS. Its distinguishing feature is that its generator model is T5-based. The choice of this model seems more suitable

[15] https://huggingface.co/t5-base

for this task, because gap filling is the main pre-training objective of T5, whereas BART has been pretrained to reconstruct texts with many other types of noise, such as token deletion, sentence permutation, and document rotation.

In terms of the content preserving formality transfer task, the important slots are sent to the model together with the source text. As we assume the availability of the parallel data, we get the list of important slots from the words of the target text that are similar to the ones in the source text. However, the first part of the LEWIT pipeline (token tagger) can also generate labels indicating which tokens should be preserved. Thus, we try combinations of the templates used for inference of the trained generator model: from predefined slots only, like shown on the bottom part of Figure 2, (*LEWIT-constr*) and from predefined slots and tagger labels (*LEWIT-constr-tag*).

We perform additional experiments that do not assume the availability of predefined slots. The templates for these approaches are obtained from the aforementioned RoBERTa-based tagger labels (*LEWIT-tag*) and third-party NER-tagger (*LEWIT-NER*). A significant part of the tags generated with the tagger within the test set was either "replace" (46,7%) or "equal" (50%). "Delete" and "insert" took 3% and 0.3% correspondingly. This proportion corresponds to the general intuition of small-edits-based paraphrasing of formal texts into more informal style by keeping the most important content intact and either slightly altering or sometimes deleting less important parts.

6 Results

The results are grouped according to the availability of the predefined important slots or constraints in the inference time and are shown in Table 2. We can see that both naive approaches are pushed to the bottom of the tables and their joint measure value is substantially less than the closest non-naive approach. We can also see that the LEWIT approach outperforms all strong baselines in both settings of the experiments.

Case 1: the slots are not known in the inference time. Both NER-tagger and edits-tagger-based approaches perform similarly by the joint measure outperforming the baseline methods. The edits-tagger approach yields better content and slots preservation but worse style transfer accuracy. The examples of the generated paraphrases are shown in Table 3.

Case 2: the slots are known in the inference time. Different variations of LEWIT also outperform the baseline methods. We can see that if the important slots are known, their combination with the edits token tagger can increase content preservation, however, this yields a decrease in style accuracy. The examples of the generated paraphrases are shown in Table 4.

The examples suggest that pure seq2seq models (such as *seq2seq-t5-para*) occasionally change the overall intent or specific slots in undesirable ways, and simple approaches to slot preservation (such as *delex*) sometimes result in unnatural outputs. Edit-based methods seem to avoid these problems in most cases.

Method	Style Accuracy	Content Preservation	Slot Preservation	Fluency	Joint
Without known constraints					
LEWIT-tag (T5)	0.69	0.85	0.98	0.75*	**0.43**
LEWIT-NER (T5)	0.82	0.74	0.94	0.74	0.42
LEWIS-tag [19] (BART)	0.77	0.69	0.99	0.72	0.38
seq2seq-t5-para	0.60	0.82	0.95	0.74	0.34
seq2seq-t5	0.54	0.87	0.98	0.73	0.33
rerank-BLEURT	0.46	0.84	0.97	0.75*	0.28
GPT2-tuned [20]	**0.91**	0.57	0.76	0.69	0.27
copy-paste	0.03	**0.90**	**1.00**	**0.83**	0.02
With known constraints					
LEWIT-constr (T5)	0.80	0.76	1.00	0.76*	**0.46**
LEWIT-constr-tag (T5)	0.73	**0.83**	1.00	0.74	0.45
LEWIS-constr [19] (BART)	0.81	0.68	1.00	0.75	0.41
delex [5]	0.69	0.75	0.98	0.78*	0.40
seq2seq-t5-template	0.49	0.83	0.97	0.78*	0.31
seq2seq-t5-constr [7]	0.71	0.71	1.00	0.61	0.31
slot-tags [27]	0.56	0.74	0.97	0.76	0.31
seq2seq-t5-para-constr	0.64	0.74	1.00	0.61	0.29
rerank-BLEURT-constr	0.54	0.77	1.00	0.64	0.27
GPT2-tuned-constr [20]	**0.91**	0.41	0.82	0.68	0.21
only-slots	0.73	0.21	1.00	**0.81**	0.12

Table 2. Results with and without the usage of the predefined important slots. "Joint" is the average product of all four measures. The values **in bold** show the highest value of the metric with the significance level of $\alpha = 0.05$ (by Wilcoxon signed-rank test). The values with an insignificant difference between LEWIT and other methods are marked with a "*" sign. The highest value for the slot preservation for the methods with known constraints is not indicated because in most cases all constraints are preserved by the design of the methods from this group.

LEWIS-based models demonstrate top performance within the baselines, however, LEWIT still performs better according to the joint score. This is most probably an evidence that T5 fits better than the BART model for the specific task of gap filling.

Results in Table 2 pass Wilcoxon signed-rank test [25] with a significance threshold of 0.05. We tested the hypothesis of the significance of the difference between the best-performing LEWIT method within each group and all baseline methods. The test was performed by splitting the test set into 30 random parts of 900 samples and calculating the significance over the mean of the measurement values from the selected samples.

Certainly, LEWIT has its limitations. The most notable one is that sticking to the structure of the source sentence limits the ability of the TST model to alter its syntactical structure. Moreover, in some contexts, a text may look more natural if rewritten from scratch.

We see the main use-case of LEWIT in applications where exact preservation of content is crucial, such as goal-oriented dialogue systems (e.g. pizza ordering), where communication goals cannot be compromised for better fluency.

Formal source text	Important slots	Informal rewrite	System
SAN International Airport is the location of flight departure.	SAN International Airport	SAN International Airport is where the flight departs	LEWIT-constr
		SAN International Airport International airport is the place of flight departure.	LEWIS-constr
		SAN International Airport is the start of your flight.	delex
How will the weather be in Delhi, India on the tomorrow?	Delhi; India; tomorrow	whats the weather like in Delhi, India for tomorrow .	LEWIT-constr
		Will the weather be in india on the next day?	LEWIS-constr
		i'm not sure, but it's going to be cool in Delhi, India on tomorrow	delex
Your destination will be at Sacramento Valley Station.	Sacramento Valley Station	go to Sacramento Valley Station !	LEWIT-constr
		destination will be at Sacramento Valley Station	LEWIS-constr
		Sacramento Valley Station	delex

Table 3. Examples of samples generated by top performing formality transfer systems with known constraints.

Formal source text	Informal rewrite	System
No I am leaving on the 3rd from Seattle, WA.	No I 'm leaving on the 3rd from Seattle, WA	LEWIT-tag
	No I leaving on the 3rd from Seattle, WA	LEWIS-tag
	No I'm leaving on the 3rd from Seattle WA.	seq2seq-t5-para
Do you have any preference in city and type of events, for example, music or Sports something like that?	do you like city or music or Sports something like that?	LEWIT-tag
	u like city and sports like that music or sports something like that?	LEWIS-tag
	Do you like city and type of things, like music or sports?	seq2seq-t5-para
I will be returning Tuesday next week.	I 'll be back Tuesday next week !	LEWIT-tag
	I l be back tUESDAY next week.	LEWIS-tag
	I'll be back on Tuesday next week.	seq2seq-t5-para
I would like to leave tomorrow from Atlanta.	I 'm leaving tomorrow from Atlanta.	LEWIT-tag
	I to get away from Atlanta.tomorrow	LEWIS-tag
	i want to leave tomorrow from atlanta	seq2seq-t5-para

Table 4. Examples of samples generated by top performing formality transfer systems without known constraints.

7 Conclusions

In this paper, we study the ways of supervised transfer of formal text to more informal paraphrases with special attention to preserving the content. In this task, the content of the source text is supposed to have a set of predefined important slots that should be kept in the generated text in either their original or slightly changed form but without a change of their meaning. To evaluate various methods for this task we collect a dataset of parallel formal-informal texts all of which have a set of predefined important slots. Using the new dataset we perform a computational study of modern approaches to supervised style transfer in two settings: with and without information about the predefined important slots provided at the inference time.

Results of our study show that if content preservation is a crucial goal, methods that do not rewrite the text completely are preferable. In this setting, it is better to use a token tagger marking spans with key information to be kept (named entities, etc.) from everything else which can be rewritten more freely with a separate generator that rephrases the rest. We show that the LEWIS [10] approach operating in this way outperforms strong baselines trained on parallel data by a large margin. We also show the original model can be substantially further improved if the T5-based generator is used.

Acknowledgements

This work was supported by the MTS-Skoltech laboratory on AI, the European Union's Horizon 2020 research and innovation program under the Marie Skodowska-Curie grant agreement No 860621, and the Galician Ministry of Culture, Education, Professional Training, and University and the European Regional Development Fund (ERDF/FEDER program) under grant ED431G2019/04.

References

1. Babakov, N., Dale, D., Logacheva, V., Krotova, I., Panchenko, A.: Studying the role of named entities for content preservation in text style transfer. In: Rosso, P., Basile, V., Martínez, R., Métais, E., Meziane, F. (eds.) Natural Language Processing and Information Systems. pp. 437–448. Springer International Publishing, Cham (2022)
2. Babakov, N., Dale, D., Logacheva, V., Panchenko, A.: A large-scale computational study of content preservation measures for text style transfer and paraphrase generation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. pp. 300–321. Association for Computational Linguistics, Dublin, Ireland (May 2022)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
4. Chen, G., Chen, Y., Wang, Y., Li, V.O.: Lexical-constraint-aware neural machine translation via data augmentation. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. pp. 3587–3593. International Joint Conferences on Artificial Intelligence Organization (7 2020), main track

5. Cui, R., Agrawal, G., Ramnath, R.: Constraint-embedded paraphrase generation for commercial tweets. In: Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. p. 369–376. ASONAM '21, Association for Computing Machinery, New York, NY, USA (2021)
6. Gu, J., Wang, C., Junbo, J.Z.: Levenshtein Transformer. Curran Associates Inc., Red Hook, NY, USA (2019)
7. Hu, J.E., Khayrallah, H., Culkin, R., Xia, P., Chen, T., Post, M., Van Durme, B.: Improved lexically constrained decoding for translation and monolingual rewriting. 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). pp. 839–850. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
8. Jhamtani, H., Gangal, V., Hovy, E., Nyberg, E.: Shakespearizing modern language using copy-enriched sequence to sequence models. In: Proceedings of the Workshop on Stylistic Variation. pp. 10–19. Association for Computational Linguistics, Copenhagen, Denmark (2017)
9. Krishna, K., Wieting, J., Iyyer, M.: Reformulating unsupervised style transfer as paraphrase generation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 737–762. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.55
10. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020)
11. Li, H., Huang, G., Cai, D., Liu, L.: Neural machine translation with noisy lexical constraints. IEEE/ACM Transactions on Audio, Speech, and Language Processing **28**, 1864–1874 (2020). https://doi.org/10.1109/TASLP.2020.2999724
12. Luo, F., Li, P., Yang, P., Zhou, J., Tan, Y., Chang, B., Sui, Z., Sun, X.: Towards fine-grained text sentiment transfer. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2013–2022. Association for Computational Linguistics, Florence, Italy (Jul 2019)
13. Moskovskiy, D., Dementieva, D., Panchenko, A.: Exploring cross-lingual text detoxification with large multilingual language models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. pp. 346–354. Association for Computational Linguistics, Dublin, Ireland (May 2022)
14. Pavlick, E., Tetreault, J.: An empirical analysis of formality in online communication. Transactions of the Association for Computational Linguistics **4**, 61–74 (2016)
15. Popović, M.: chrF: character n-gram F-score for automatic MT evaluation. In: Proceedings of the Tenth Workshop on Statistical Machine Translation. pp. 392–395. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015). https://doi.org/10.18653/v1/W15-3049
16. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(1) (jan 2020)
17. Rao, S., Tetreault, J.: 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). pp. 129–140. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018)
18. Rastogi, A., Zang, X., Sunkara, S., Gupta, R., Khaitan, P.: Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 8689–8696 (2020)
19. Reid, M., Zhong, V.: LEWIS: Levenshtein editing for unsupervised text style transfer. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 3932–3944. Association for Computational Linguistics, Online (Aug 2021)
20. Reif, E., Ippolito, D., Yuan, A., Coenen, A., Callison-Burch, C., Wei, J.: A recipe for arbitrary text style transfer with large language models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 837–848. Association for Computational Linguistics, Dublin, Ireland (May 2022)
21. Sellam, T., Das, D., Parikh, A.: BLEURT: Learning robust metrics for text generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7881–7892. Association for Computational Linguistics, Online (Jul 2020)
22. Shen, T., Lei, T., Barzilay, R., Jaakkola, T.: Style transfer from non-parallel text by cross-alignment. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2017)
24. Warstadt, A., Singh, A., Bowman, S.R.: Neural network acceptability judgments. arXiv preprint arXiv:1805.12471 (2018)
25. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin **1**(6), 80–83 (1945)
26. Wu, X., Zhang, T., Zang, L., Han, J., Hu, S.: Mask and infill: Applying masked language model for sentiment transfer. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. pp. 5271–5277. International Joint Conferences on Artificial Intelligence Organization (7 2019)
27. Zhang, M., Tan, L., Fu, Z., Xiong, K., Lin, J., Li, M., Tu, Z.: Don't change me! user-controllable selective paraphrase generation. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. pp. 3522–3527. Association for Computational Linguistics, Online (Apr 2021)
28. Zhang, Y., Ge, T., Sun, X.: Parallel data augmentation for formality style transfer. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 3221–3228. Association for Computational Linguistics, Online (Jul 2020)
29. Zhang, Y., Wang, G., Li, C., Gan, Z., Brockett, C., Dolan, B.: POINTER: Constrained progressive text generation via insertion-based generative pre-training. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 8649–8670. Association for Computational Linguistics, Online (Nov 2020)
30. Zhuang, L., Wayne, L., Ya, S., Jun, Z.: A robustly optimized BERT pre-training approach with post-training. In: Proceedings of the 20th Chinese National Conference on Computational Linguistics. pp. 1218–1227. Chinese Information Processing Society of China, Huhhot, China (Aug 2021)