Semi-Supervised Formality Style Transfer with Consistency Training

Ao Liu, An Wang, Naoaki Okazaki
Tokyo Institute of Technology
liu.ao@nlp.c.titech.ac.jp
wang@de.cs.titech.ac.jp
okazaki@c.titech.ac.jp

Abstract

Formality style transfer (FST) is a task that involves paraphrasing an informal sentence into a formal one without altering its meaning. To address the data-scarcity problem of existing parallel datasets, previous studies tend to adopt a cycle-reconstruction scheme to utilize additional unlabeled data, where the FST model mainly benefits from target-side unlabeled sentences. In this work, we propose a simple yet effective semi-supervised framework to better utilize source-side unlabeled sentences based on consistency training. Specifically, our approach augments pseudo-parallel data obtained from a source-side informal sentence by enforcing the model to generate similar outputs for its perturbed version. Moreover, we empirically examined the effects of various data perturbation methods and propose effective data filtering strategies to improve our framework. Experimental results on the GYAFC benchmark demonstrate that our approach can achieve state-of-the-art results, even with less than 40% of the parallel data[1].

1 Introduction

Formality style transfer (FST) (Rao and Tetreault, 2018) has garnered growing attention in the text style transfer community, which aims to transform an *informal*-style sentence into a *formal* one while preserving its meaning. The large amount of user-generated data from online resources like tweets often contain informal expressions such as slang words (e.g., *gonna*), wrong capitalization or punctuations, and grammatical or spelling errors. FST can clean and formalize such noisy data, to benefit downstream NLP applications such as sentiment classification (Yao and Yu, 2021a). Some examples of FST data are presented in Table 1.

With the release of the FST benchmark Grammarly Yahoo Answers Corpus (GYAFC) (Rao and Tetreault, 2018), previous studies on FST tend to employ neural networks such as sequence-to-sequence (seq2seq) models to utilize parallel (informal and formal) sentence pairs. However, GYAFC only contains 100k parallel examples, which limits the performance of neural network models. Several approaches have been developed to address the data-scarcity problem by utilizing unlabeled sentences. In a previous study, Zhang et al. (2020) proposed several effective data augmentations methods, such as back-translation, to augment parallel data. Another line of research (Shang et al., 2019; Xu et al., 2019; Chawla and Yang, 2020) conducted semi-supervised learning (SSL) in a cycle-reconstruction manner, where both forward and backward transfer models were jointly trained while benefiting each other by generating pseudo-parallel data from unlabeled sentences. Under this setting, both additional informal and formal sentences are utilized; however, the forward informal→formal model mostly benefits from the *target*-side (formal) sentences, which are back-translated by the formal→informal model to construct pseudo training pairs. Conversely, the formal→informal model can only acquire extra supervision signals from informal sentences. Because the main objective of FST is the informal→formal transfer, the additional informal sentences were not well utilized in previous studies. In addition, these semi-supervised models incorporate many auxiliary modules such as style discriminators, to achieve state-of-the-art results, which result in rather complicated frameworks and more model parameters.

Informal	*TITANIC I THINK IT COST ABOUT 300 MILLION*
Formal	*I think that Titanic cost around 300 million dollars.*
Informal	*being condiderate of her feelings and needs*
Formal	*I am being considerate of her personal needs and feelings.*

Table 1: Examples of informal-formal sentence pairs.

As noisy informal sentences are easier to ac-

[1]Code available at https://github.com/Aolius/semi-fst.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4689 - 4701

quire from online resources, we attempt to take a different view from existing approaches, by adopting additional *source*-side (informal) sentences via SSL. We gain insights from the state-of-the-art approaches for semi-supervised image and text classification (Sohn et al., 2020; Xie et al., 2020; Berthelot et al., 2019; Zhang et al., 2021; Chen et al., 2020) and propose a simple yet effective SSL framework for FST using purely informal sentences. Our approach employs *consistency training* to generate pseudo-parallel data from additional informal sentences. Specifically, we enforce the model to generate similar target sentences for an unlabeled source-side sentence and its perturbed version, making the model more robust against the noise in the unlabeled data. In addition, a supervised loss is trained simultaneously to transfer knowledge from the clean parallel data to the unsupervised consistency training.

Data perturbation is the key component of consistency training and significantly affects its performance. To obtain a successful SSL framework for FST, we first empirically study the effects of various data perturbation approaches. Specifically, we explore easy data augmentation methods, such as random *word deletion*, and advanced data augmentation methods, such as *back-translation*. We also handcraft a line of rule-based data perturbation methods to simulate the features of informal sentences, such as *spelling error injection*. Furthermore, we propose three data filtering approaches in connection with the three evaluation metrics of FST: style strength, content preservation, and fluency. Specifically, we adopt *style accuracy*, *source*-BLEU, and *perplexity* as three metrics to filter out low-quality pseudo-parallel data based on a threshold. We also propose a dynamic threshold algorithm to automatically select and update the thresholds of source-BLEU and perplexity.

We evaluate our framework on the two domains of the GYAFC benchmark: *Entertainment & Music* (E&M) and *Family & Relationships* (F&R). We further collect 200k unpaired informal sentences for each domain to perform semi-supervised training. Experimental results verify that our SSL framework can enhance the performance of the strong supervised baseline, a pretrained T5-large (Raffel et al., 2020) model, by a substantial margin, and improve the state-of-the-art results by over 2.0 BLEU scores on both GYAFC domains. Empirically, we also deduce that simple word-level data augmentation approaches are better than advanced data augmentation methods that excessively alter the sentences, and *spelling error injection* is especially effective. In addition, our evaluation-based data filtering approach can further improve the performance of the SSL framework. Furthermore, we also conduct low-resource experiments by reducing the size of parallel data. Surprisingly, our framework could achieve the state-of-the-art results with only less than 40% of parallel data, demonstrating the advantage of our method in low-resource situations.

2 Related Work

Formality style transfer FST is an important branch of text style transfer. For FST, Rao and Tetreault (2018) released a high-quality parallel dataset - GYAFC, comprising two sub-domains and approximately 50k parallel data for each domain. Previous studies (Rao and Tetreault, 2018; Niu et al., 2018; Xu et al., 2019; Zhang et al., 2020) typically train seq2seq encoder-decoder models on this benchmark. Recent studies (Wang et al., 2019; Yao and Yu, 2021b; Chawla and Yang, 2020; Lai et al., 2021) have deduced that fine-tuning large-scale pretrained models such as GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2020) on the parallel corpora can improve the performance. To address the data-scarcity problem of parallel datasets, Zhang et al. (2020) proposed three data augmentation techniques to augment pseudo-parallel data for training. Similar to prior research on text style transfer that adopt back-translation (Zhang et al., 2018; Lample et al., 2018; Prabhumoye et al., 2018; Luo et al., 2019), some other approaches on FST (Shang et al., 2019; Xu et al., 2019; Chawla and Yang, 2020) adopt a cycle-reconstruction scheme, where an additional backward transfer model is jointly trained together with the forward transfer model, and the two models generate pseudo-paired data for each other via iterative back-translation. Although Xu et al. (2019) and Chawla and Yang (2020) train a single model to perform bidirectional transfer, the generation of both directions remain disentangled by a control variable, making each direction rely on the unlabeled data of its target side. Therefore, the unlabeled informal sentences exert no direct effects on the informal→formal transfer. In contrast, our work focuses on how to better utilize source-side unlabeled data (i.e., informal sentences) using SSL and does not introduce any extra models.

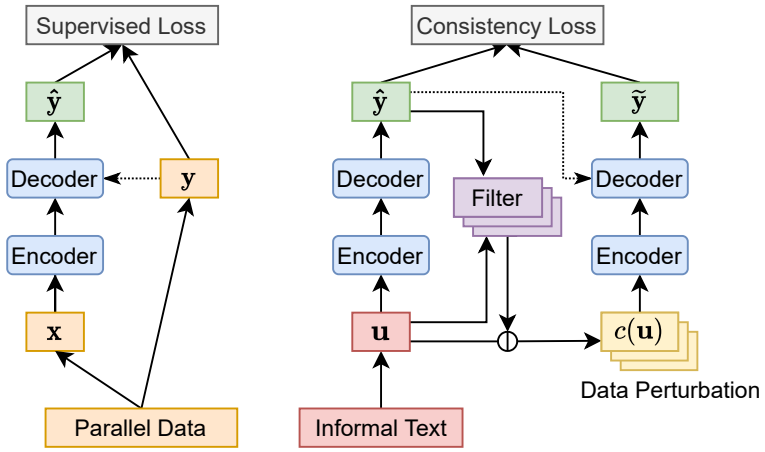

Figure 1: Overview of our semi-supervised consistency training framework which jointly optimizes two losses: (1) a supervised loss trained on parallel data; (2) a consistency training loss, where the model first generates a pseudo target $\hat{\mathbf{y}}$ for an additional informal text, then it is combined with a perturbed input $c(\mathbf{u})$ to train the model if passing the data filter. The dotted arrows indicate teacher forcing in the encoder-decoder model.

SSL with consistency regularization SSL is popular for its advantage in utilizing unlabeled data. Consistency regularization (also known as consistency training) (Sajjadi et al., 2016) is an important component of recent SSL algorithms on image and text classification (Miyato et al., 2018; Tarvainen and Valpola, 2017; Berthelot et al., 2019; Sohn et al., 2020). It enforces a model to produce invariant predictions for an unlabeled data and its perturbed version. These studies developed different data perturbation (Xie et al., 2020; Berthelot et al., 2019) or data filtering (Zhang et al., 2021; Xu et al., 2021) approaches to improve the performance. However, few studies have been made on how to apply consistency training in natural language generation (NLG) tasks such as FST because of the different target spaces, i.e., instead of single class labels or probabilities, the output of NLG is the combination of discrete NL tokens. This renders the experiences in classification tasks not applicable to FST. For instance, classification probabilities are typically adopted as the metric to filter high-confidence pseudo-examples for consistency training in classification tasks (Sohn et al., 2020; Xie et al., 2020; Zhang et al., 2021), which is implausible in FST. A similar study (He et al., 2019) improved self-training by injecting noise into unlabeled inputs and proved its effectiveness on machine translation and text summarization; however, self-training involves multiple iterations to collect pseudo-parallel data and retrain the model, hence the training is not end-to-end. In this study, we explore various data perturbation strategies and propose effective data filtering approaches to realize a successful consistency-based framework for FST, which may also provide useful insights for future studies on semi-supervised NLG.

3 Method

3.1 Base Model

FST involves rewriting an informal sentence into a formal one. Formally, given a sentence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ of length n with style S, our objective is to transform it into a target sentence $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ of length m and style T, while preserving its content.

Following prior studies (Rao and Tetreault, 2018; Zhang et al., 2020; Chawla and Yang, 2020; Lai et al., 2021) on FST, we employ the supervised baseline as a seq2seq encoder-decoder model that directly learns the conditional probability $P(\mathbf{y}|\mathbf{x})$ from parallel corpus $\mathcal{D}$ comprising $(\mathbf{x}, \mathbf{y})$ pairs. The objective is the cross-entropy loss between the decoder outputs and the ground-truth target sentences:

$$\begin{aligned}\mathcal{L}_{sup} &= \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim D}[-\log P(\mathbf{y}|\mathbf{x};\theta)] \\ &= \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim D}[-\sum_i \log P(y_i|y_{1:i-1},\mathbf{x};\theta)],\end{aligned} \tag{1}$$

where θ denotes the model parameters.

3.2 Consistency Training

Our approach leverages the idea of consistency regularization (Sajjadi et al., 2016) and enforces a model to generate similar target sentences for an

original and perturbed unlabeled sentence. Simultaneously, the model is also trained on the supervised data. Accordingly, the knowledge garnered from supervised training can be gradually transferred to unsupervised training. An overview of our framework is presented in Figure 1. Typically, the consistency training loss is computed on the divergence between predictions on an unlabeled input $\mathbf{u}$ and its perturbed version $\tilde{\mathbf{u}} = c(\mathbf{u})$, where $c(\cdot)$ is the perturbation function and $\mathbf{u} \in \mathcal{U}_S$ represents a source-side unlabeled sentence (in our case, an informal sentence). Formally, consistency training can be defined as minimizing the following unsupervised loss:

$$\mathbb{E}_{\mathbf{u}\sim\mathcal{U}_S} D\left[P(\mathbf{y}|\mathbf{u};\theta)||P(\mathbf{y}|c(\mathbf{u});\theta)\right], \quad (2)$$

where $D[\cdot||\cdot]$ denotes a divergence loss. In practice, we adopt pseudo-labeling (Lee et al., 2013) to train the unsupervised loss, for which we fix the model parameter θ to predict a "hard label" (pseudo target sentence) $\hat{\mathbf{y}}$ for $\mathbf{u}$ and enforce the consistency of model prediction by training θ with $(c(\mathbf{u}), \hat{\mathbf{y}})$. Hence the unsupervised objective can be optimized as a standard cross-entropy loss as follows:

$$\mathcal{L}_{unsup} = \mathbb{E}_{\mathbf{u}\sim\mathcal{U}_S}\mathbb{E}_{\hat{\mathbf{y}}\sim P(\mathbf{y}|\mathbf{u};\hat{\theta})}[-\log P(\hat{\mathbf{y}}|c(\mathbf{u});\theta)], \quad (3)$$

where $\hat{\theta}$ denotes a fixed copy of θ. This training process does not introduce additional model parameters. The entire additional training cost to supervised learning is a training pass and a generation pass for each unlabeled sentence.

As the overall objective, we train a weighted sum of the supervised loss in Equation (1) and the unsupervised loss in Equation (3):

$$\mathcal{L} = \mathcal{L}_{sup} + \lambda\mathcal{L}_{unsup}, \quad (4)$$

where λ represents a hyper-parameter for balancing the effects of supervised and unsupervised training. To achieve a good initial model for consistency training, we first pretrain the model on the supervised loss for several warm-up steps.

3.3 Data Perturbation Strategies

Data perturbation is the key component of consistency-based SSL algorithms (Xie et al., 2020; Chen et al., 2020) and significantly affects the performance. In this section, we briefly introduce a collection of different data perturbation methods explored in this research.

First, we consider some easy data augmentation methods commonly used for supervised data augmentation, which includes

- **word deletion (drop)**[2]: to randomly drop a proportion of words in the sentence.
- **word swapping (swap)**: to randomly swap a proportion of words with their neighbouring words.
- **word masking (mask)**: to randomly replace words with a mask token "_".
- **word replacing with synonym (synonym)**: to randomly replace some words with a synonym based on WordNet (Fellbaum, 1998).

In addition, we consider advanced data augmentation methods that have proven effective in semi-supervised text classification (Xie et al., 2020):

- **back-translation**: to translate a sentence into a pivot language, then translate it back to obtain a paraphrase of the original one.
- **TF-IDF based word replacing (tf-idf)**: to replace uninformative words with low TF-IDF scores while retaining those with high TF-IDF values.

Furthermore, we handcraft a set of rule-based data perturbation for FST. There are some typical informal expressions in the parallel corpus, such as the use of slang words and abbreviations, capitalized words for emphasis, and spelling errors. Some existing studies (Wang et al., 2019; Yao and Yu, 2021b) adopt editing rules to revise such informal expressions as a preprocessing step. Inspired by these, we propose the adoption of opposite rules to synthesize such noises. We consider the following methods:

- **spelling error injection (spell)**: to randomly inject spelling errors to a proportion of words by referring to a spelling error dictionary.
- **word replacing with abbreviations (abbr)**: to replace all the words in the sentence with their abbreviations or slang words (e.g., "are you" → "r u") by referring to an abbreviation dictionary.
- **word capitalization (capital)**: to randomly capitalize a proportion of words.

[2]We abbreviate each method for ease of denotation.

These rule-based methods can inject noise into the unlabeled informal sentences without changing its informality, but strengthening it instead.

3.4 Evaluation-Based Data Filtering

In the consistency training loss, the noisy pseudo-target $\hat{\mathbf{y}}$ is generated from the decoder model and may exert negative effects on the training. Therefore, we propose three evaluation-based data filters in connection with the evaluation metrics of FST.

Specifically, we attempt to measure the quality of pseudo-target sentences by considering the three most important evaluation criteria of text style transfer: **style strength**, **content preservation**, and **fluency**. Next, we comprehensively explain each evaluation metric and the corresponding data filter.

Style strength measures the formality of generated sentences. Typically, people adopt binary classifiers such as TextCNN (Chen, 2015) classifiers to judge the formality of a sentence (Lai et al., 2021). Inspired by this, we pretrain a TextCNN formality classifier on the parallel training corpus (i.e., GYAFC) to distinguish between informal and formal sentences. For an unlabeled informal sentence $\mathbf{u}$ and its pseudo target sentence $\hat{\mathbf{y}}$, we maintain $(c(\mathbf{u}), \hat{\mathbf{y}})$ for unsupervised training only when

$$p^{+}_{cls}(\hat{\mathbf{y}}) - p^{+}_{cls}(\mathbf{u}) > \sigma, \tag{5}$$

where $p^{+}_{cls}(\cdot)$ represents the probability of the sentence being formal, predicted by the style classifier and σ is a threshold of the probability. This guarantees that only the sentence pairs with strong style-differences are used for consistency training.

Content preservation is another important evaluation metric of FST, typically measured with BLEU between the ground-truth target sentence and the model generations. In unsupervised text style transfer where no ground-truth target exists, *source*-BLEU is adopted as an alternative, i.e., the BLEU scores between the source input sentence and the generated target sentence. Similarly, we propose the adoption of *source*-BLEU between $\mathbf{u}$ and $\hat{\mathbf{y}}$ as the metric to filter out pseudo targets that present poor content preservation.

Fluency is also used to evaluate the quality of generated sentences. We follow (Hu et al., 2020) to pretrain an N-gram language model on the training data to estimate the empirical distributions of formal sentences. Then, the *perplexity* score is calculated for the pseudo target sentence $\hat{\mathbf{y}}$ by the language model. The motivation is that the sentences with lower perplexity scores match the empirical distribution of formal sentences better, and are thus considered as more fluent.

A natural idea is to filter out pseudo-parallel data based on a *source*-BLEU or a *perplexity* threshold. However, it is infeasible to determine the optimal threshold for the two metrics beforehand because the pseudo paired data are generated on-the-fly during the training and we cannot know the distribution of the BLEU or perplexity scores. In addition, choosing the BLEU/perplexity threshold is not as easy as tuning the style probability σ because they heavily depend on the data distribution and exhibit varying ranges of values.

3.5 Dynamic Threshold Selection

To realize the selection of thresholds for the BLEU- and perplexity- based filters, we propose a dynamic threshold strategy based on the distribution of the scores computed for already generated pseudo-paired sentences. Specifically, we maintain an ordered list L to store the scores calculated for previously generated pseudo data and update it continuously following the training. At each iteration, a batch of new scores are inserted into L while maintaining the decreasing order of the list. Subsequently, we update the threshold as the value at a certain position $L[\phi \times len(L)]$ in the score list, where $len(L)$ denotes the length of the current score list and $\phi \in [0, 1]$ represents a ratio that determines the threshold's position in the list. We only keep pseudo data with scores higher (or lower for perplexity scores) than the threshold for consistency training. This actually makes ϕ approximately the proportion of pseudo data we keep for training, making it more convenient to control the trade-off between the qualities and quantities of selected pseudo data. More details are provided in Appendix B, C.

4 Experiments

We introduce the experimental settings in Section 4.1. To obtain relevant findings on how to build an effective consistency training framework for FST, we first empirically study the effects of multiple data perturbation methods in Section 4.2 and prove the effectiveness of consistency training via comparisons with the base model. Then, we validate our consistency training model with different data filtering methods in Section 4.3 and demonstrate their additional effects on the SSL frame-

Dataset	Train	Val	Test	Unlabeled
E&M	52595	2877	1416	200k
F&R	51967	2788	1432	200k

Table 2: The statistics of datasets.

work. Based on the findings in these two experiments, we further compare our best models with previous state-of-the-art models in Section 4.4. We also include case studies in Section 4.4 to present some qualitative examples. Finally, we conduct low-resource experiments (Section 4.5) to demonstrate our method's advantage when less parallel data are available.

4.1 Experimental Settings

Datasets We evaluate our framework on the GYAFC (Rao and Tetreault, 2018) benchmark for formality style transfer. It comprises crowdsourced informal-formal sentence pairs split into two domains, namely, E&M and F&R. The informal sentences in the dataset were originally selected from the same domains in Yahoo Answers L6 corpus[3]. We focus on the *informal-formal* style transfer because it is more realistic in applications. We further collected massive amounts of informal sentences from each of the two domains in Yahoo Answers L6 corpus as the unsupervised data. The statistics of the datasets are presented in Table 2.

Implementation Details We employ PyTorch (Paszke et al., 2019) for all the experiments. We pretrain a TextCNN style classifier on the supervised data for each domain of GYAFC, following the setting in (Lai et al., 2021). The same classifier is adopted for both the style accuracy evaluation and the style strength filter in our SSL framework. We adopt HuggingFace Transformers (Wolf et al., 2020) library's implementation of pretrained T5-Large (Raffel et al., 2020) as the base model. We adopt the Adam (Kingma and Ba, 2014) optimizer with the initial learning rate 2×10^{-5} to train all the models. More details of hyper-parameters and model configurations are provided in Appendix A.

Evaluation Metrics The main evaluation metric for FST is the BLEU score between the generated sentence and four human references in the test set. We adopt the corpus BLEU in NLTK (Loper and Bird, 2002) following (Chawla and Yang, 2020). In addition, we also pretrained a TextCNN formality classifier to predict the formality of transferred sentences and calculate the accuracy (Acc.). Furthermore, we compute the harmonic mean of BLEU and style accuracy as an overall score, following the settings in (Lai et al., 2021).

[3]https://webscope.sandbox.yahoo.com/catalog.php?datatype=l

4.2 Effects of Data Perturbation Methods

In this experiment, we validate the effectiveness of our consistency training framework and compare the effects of different data perturbation methods. Specifically, we adopt the nine data perturbation methods introduced in Section 3.3 and include the *no-perturbation* variant that indicates directly using an unlabeled sentence and its pseudo target to train the unsupervised loss. We adopted no data filtering strategy in this experiment to simplify the comparison.

As shown in Table 3, our framework could consistently improve the base model by using different perturbation methods; however, back-translation resulted in mostly lower results than the base model. This contradicts the conclusion in (Xie et al., 2020) that back-translation is especially powerful for semi-supervised text classification. We attribute this to the fact that back-translation tends to change the entire sentence into a semantically similar but syntactically different sentence. Compared with other word-level perturbation strategies, back-translation triggers a larger mismatch between the perturbed input and the pseudo-target sentence generated from the unperturbed input, leading to a poor content preservation ability of the model. In contrast, simple word-level noises achieved consistently better results, especially spell error (*spell*), random word swapping (*swap*), and abbreviation replacing (*abbr*). These three methods tend to alter the words but do not lose their information while other methods eliminate the entire word by deleting (*drop, mask*) or replacing it with another word (*synonym, tf-idf*). This may also cause a larger mismatch between the pseudo input and output.

Hence, we draw the conclusion that *simple word-level perturbations tend to bring more effects*. This differs from the observations in text classification (Xie et al., 2020) because content preservation is important in FST. In particular, we also found that *spell* achieved the highest BLEU scores on both datasets. However, adding no perturbation even resulted in a worse performance than the base model. Moreover, *capital* is also relatively weaker than the other two rule-based methods because it

	E&M			F&R		
Model variants	BLEU	Acc(%)	HM	BLEU	Acc(%)	HM
base model	76.87	90.04	82.94	80.32	84.01	82.12
no-perturbation	76.41	88.49	82.01	79.22	84.46	81.75
drop	77.55	93.15	84.64	80.53	86.56	83.44
swap	77.90	93.43	84.96	81.07	85.96	83.44
mask	77.52	93.93	84.94	80.69	86.41	83.45
synonym	77.48	93.64	84.80	80.49	86.26	83.28
back-translation	76.07	90.11	82.50	79.96	84.91	82.36
tf-idf	76.89	92.58	84.01	80.48	**86.94**	83.58
abbr	77.55	93.64	84.84	81.00	**86.94**	**83.86**
capital	77.54	93.15	84.63	80.74	85.74	83.16
spell	**78.37**	**94.21**	**85.56**	**81.09**	85.59	83.28

Table 3: Effects of different data perturbations in our approach on the test splits of GYAFC. The best scores among all the model variants are boldfaced.

	E&M			F&R		
Model variants	BLEU	Acc(%)	HM	BLEU	Acc(%)	HM
spell (no-filter)	78.37	94.21	85.56	81.09	85.59	83.28
spell (+style)	78.19	93.79	85.28	**81.37**	**86.41**	**83.81**
spell (+bleu)	**78.75**	**94.56**	**85.94**	**81.11**	**86.34**	**83.64**
spell (+lm)	78.24	**94.56**	**85.63**	80.93	**86.34**	**83.55**

Table 4: Effects of different data filtering methods in our approach on the test splits of GYAFC. Scores larger than the no-filter variant are in **bold**.

only changes the case of a chosen word. This suggests that the perturbation should not be too simple either.

4.3 Effects of Data Filtering

In this section, we analyze whether our proposed data filters are beneficial to the performance of our consistency training framework. Specifically, we chose the most effective data perturbation method *spell* to analyze the effects of adding the three data filters: style strength (*style*), content preservation (*bleu*), and fluency (*lm*) filters. As presented in Table 4, the results for different datasets and different filters have different tendencies. For example, adding the *style* filter on the E&M dataset caused negative effects while contributing the best results to the F&R domain.

Although a filter does not necessarily improve the result, this is reasonable because filters result in less pseudo data for model training and it is difficult to control the trade-off between the quality and the quantity of selected data. Nevertheless, we still observe that the *bleu* filter contributes to the highest performance of *spell* for all the metrics on the E&M domain, while *style* benefits the performance of *spell* the most on F&R, leading to the best performing models of our approach[4].

4.4 Comparison with Previous Works

We compare our best model with the following previous studies on GYAFC.

- **NMT** (Rao and Tetreault, 2018) is an LSTM-based encoder-decoder model with attention.
- **GPT-CAT** (Wang et al., 2019) adopts GPT-2 and rule-based pre-processing for informal sentences.
- **NMT-Multi-task** (Niu et al., 2018) jointly solves monolingual formality transfer and formality-sensitive machine translation via multi-task learning.
- **Hybrid Annotations** (Xu et al., 2019) trains a CNN discriminator in addition to the transfer model and adopts a cycle-reconstruction loss to utilize unsupervised data.
- **Transformers (DA)** (Zhang et al., 2020) uses three data augmentation methods, includ-

[4]Empirically, we also found that mixing up three filters achieved no better results than a single filter, possibly because this filtered out too much pseudo data.

Models	unlabeled data	E&M			F&R		
		BLEU	Acc(%)	HM	BLEU	Acc(%)	HM
NMT(Rao and Tetreault, 2018)	no	68.41	-	-	74.22	-	-
Hybrid Annotations†* (Xu et al., 2019)	yes	69.28	89.83	78.23	74.36	82.96	78.42
NMT-Multi-task† (Niu et al., 2018)	no	72.01	88.84	79.54	75.35	80.03	77.62
GPT-CAT (Wang et al., 2019)	no	71.39	-	-	77.26	-	-
Transformers (DA) (Zhang et al., 2020)	yes	74.24	-	-	77.97	-	-
CARI (Yao and Yu, 2021b)	no	74.31	-	-	78.05	-	-
Chawla's† (Chawla and Yang, 2020)	yes	76.17	91.88	83.29	79.92	83.63	81.73
BART-large+SC+BLEU† * (Lai et al., 2021)	no	76.50	94.42	84.52	79.25	**90.69**	**84.58**
Ours (base)	no	76.87	90.04	82.94	80.32	84.01	82.12
Ours (best)	yes	**78.75**	**94.56**	**85.94**	**81.37**	86.41	83.81

Table 5: Comparison between our approach and existing works on the test splits of GYAFC. † indicates we recalculate the scores with our evaluation metrics for the output given in the paper. Otherwise, we copy the results from the paper. * indicates that the model used training data from both domains and is not comparable to our model.

Model	Formality		Fluency		Meaning	
	E&M	F&R	E&M	F&R	E&M	F&R
Chawla's	1.46	1.22	1.85	1.80	**1.87**	1.88
Ours (base)	1.42	1.28	1.84	1.82	1.86	**1.95**
Ours (best)	**1.55**	**1.41**	**1.88**	**1.85**	**1.87**	1.88

Table 6: Human evaluation results.

#Parallel data	E&M		F&R	
	BLEU	Acc(%)	BLEU	Acc(%)
100 (base)	59.94	61.58	65.13	49.32
100 (ours)	**64.40**	**82.91**	**71.11**	**55.11**
1000 (base)	70.49	83.26	75.36	**76.58**
1000 (ours)	**72.22**	**85.81**	**76.70**	76.20
5000 (base)	75.13	**89.55**	77.65	78.38
5000 (ours)	**75.67**	87.08	**78.87**	**81.01**
20000 (base)	76.55	90.96	79.25	83.33
20000 (ours)	**76.59**	**92.09**	**80.61**	**86.11**

Table 7: Experimental results on test sets under low-resource settings with varied parallel data size.

ing back-translation, formality discrimination, and multi-task transfer.

- **CARI** (Yao and Yu, 2021b) improves GPT-CAT by using BERT (Devlin et al., 2018) to select optimal rules to pre-process the informal sentences.

- **Chawla's** (Chawla and Yang, 2020) uses language model discriminators and maximizing mutual information to improve a pretrained BART-Large (Lewis et al., 2020) model, along with a cycle-reconstruction loss to utilize unlabeled data.

- **BART-large+SC+BLEU** (Lai et al., 2021) improves BART-large by incorporating reinforcement learning rewards to enhance style change and content preservation.

We also report the results of **Ours (base)**, our backbone T5-large model, and **Ours (best)**, our best performing models selected from Table 4.

As observed in Table 5, **Ours (best)** outperforms previous state-of-the-art models by a substantial margin and improves the BLEU scores from 76.17 and 79.92 to 78.75 and 81.37, respectively, on the E&M and F&R domains of the GYAFC benchmark. Although **BART-large+SC+BLEU** achieved better results on the Acc. of F&R, the only released official outputs of **BART-large+SC+BLEU** were obtained from a model that was trained on the training data of both domains and adopted rewards to directly optimize style accuracy; hence, it is not directly comparable to our model. **Ours (best)** improves the fine-tuned T5-large baseline by a large margin as well, demonstrating the effectiveness of our SSL framework.

Human Evaluation We also conduct human evaluation to better capture the quality of the models' outputs. Following (Zhang et al., 2020), we measure the *Formality*, *Fluency*, and *Meaning Preservation* of generated sentences by asking two human annotators to assign a score ranging from {0, +1, +2} regarding each aspect. We randomly sampled 50 examples from the test set of each domain and compare the generated outputs of **Ours (base)**, **Ours (best)**, and the previous state-of-the-art **Chawla's** model trained on the single-domain data. In addition, the annotators were unaware of the corresponding model of each output. As shown

Example 1	Source	*I like natural / real girls, I don't like fake looking prissy drama queens.*
	Ours(best)	*I like natural looking girls, not pretentious drama queens.*
	Ours(base)	*I like natural, real girls, I do not like fake looking, prissy drama queens.*
	Chawla's	*I like natural and real girls , I do not like fake looking prissy drama queens .*
	Human-Annotation	*I like natural and real girls, not fake-looking, prissy drama queens.*
Example 2	Source	*That's like Broke Back Mountain for little John Wanye.*
	Ours(best)	*That is similar to "Broke Back Mountain" for John Wayne.*
	Ours(base)	*That is like "Broke Back Mountain" for John Wayne.*
	Chawla's	*That is like Broke Back Mountain for little John Wanye .*
	Human-Annotation	*That is similar to "Brokeback Mountain" for young John Wayne.*
Example 3	Source	*You guys don't have any reason to hate each other.*
	Ours(best)	*You do not have any reason to hate each other.*
	Ours(base)	*You guys do not have any reason to hate each other.*
	Chawla's	*You guys do not have any reason to hate each other .*
	Human-Annotation	*There is no reason for you two to dislike each other.*

Table 8: Examples sampled from the test set outputs.

in Table 6, the human evaluation results are consistent with the automatic evaluation results: **Ours (base)** is competitive compared with **Chawla's**, while **Ours (best)** improves over the base model and outperforms the previous state-of-the-art on all the metrics, except that it presents lower results on *Meaning* than **Ours (base)** on F&R. More details on human evaluation can be found in Appendix D.

Qualitative Examples We present some of the generated outputs of **Ours (base)**, **Ours (best)**, and **Chawla's** in Table 8. It can be observed that all the models can produce high-quality outputs with considerable formality, meaning preservation and fluency. Nevertheless, **Ours (best)** exhibits a stronger capability to modify the original sentence, especially for some informal expressions, leading to the best performance on the *Formality* metric. For example, it replaced "like" with "similar to" in Example 2 and deleted the informal word "guys" in Example 3. However, it may alter the original sentence so much that the meaning of the sentence is changed to some extent (Example 1). This may explain why **Ours (best)** achieves a lower *Meaning* score than **Ours (base)** on F&R.

4.5 Low-Resource Experiments

We also simulate the low-resource settings by further reducing the size of available parallel data. Specifically, we randomly sample from the original training data with a size in the range of {100, 1000, 5000, 20000} and compare the results of the base model T5-Large with our SSL model. The size of unlabeled data remains 200k for each domain. We adopt the *spell* data perturbation without any data filter and avoid exhaustive hyper-parameter tuning. Table 7 demonstrates that our framework is especially effective under few-shot settings when only 100 parallel data are available. By comparing with previous state-of-the-art results on FST, we can observe that our approach can achieve competitive results with only 5000 ($< 10\%$) parallel training data, and even better results with only 20000 ($< 40\%$) parallel examples.

5 Conclusion

In this study, we proposed a simple yet effective consistency-based semi-supervised learning framework for formality style transfer. Unlike previous studies that adopted cycle-reconstruction to utilize additional target-side sentences for back-translation, our method offers a different view, to leverage source-side unlabeled sentences. Without introducing additional model parameters, our method can easily outperform the strong supervised baseline and achieve the new state-of-the-art results on formality style transfer datasets. For future work, we will attempt to generalize our approach to other text generation scenarios.

Acknowledgements

This paper is based on results obtained from a project, JPNP18002, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). Ao Liu acknowledges financial support from the Advanced Human Resource Development Fellowship for Doctoral Students, Tokyo Institute of Technology.

References

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raf-

fel. 2019. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32:5049–5059.

Kunal Chawla and Diyi Yang. 2020. Semi-supervised formality style transfer using language model discriminator and mutual information maximization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2340–2354, Online. Association for Computational Linguistics.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Yahui Chen. 2015. Convolutional neural network for sentence classification. Master's thesis, University of Waterloo.

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*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

Zhiqiang Hu, Roy Ka-Wei Lee, Charu C Aggarwal, and Aston Zhang. 2020. Text style transfer: A review and experimental evaluation. *arXiv preprint arXiv:2010.12742*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2021. Thank you BART! rewarding pre-trained models improves formality style transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 484–494, Online. Association for Computational Linguistics.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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*, pages 7871–7880, Online. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019*.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary 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. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

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

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.

Mingyue Shang, Piji Li, Zhenxin Fu, Lidong Bing, Dongyan Zhao, Shuming Shi, and Rui Yan. 2019. Semi-supervised text style transfer: Cross projection in latent space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4937–4946.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, 30:1195–1204.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Ruochen Xu, Tao Ge, and Furu Wei. 2019. Formality style transfer with hybrid textual annotations. *arXiv preprint arXiv:1903.06353*.

Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. 2021. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pages 11525–11536. PMLR.

Zonghai Yao and Hong Yu. 2021a. Improving formality style transfer with context-aware rule injection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1561–1570, Online. Association for Computational Linguistics.

Zonghai Yao and Hong Yu. 2021b. Improving formality style transfer with context-aware rule injection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1561–1570, Online. Association for Computational Linguistics.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34.

Yi Zhang, Tao Ge, and Xu Sun. 2020. Parallel data augmentation for formality style transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3221–3228.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*.

A Detailed Experimental Settings

A.1 Hyper-Parameters

We set the max length of input sentences to 50 Byte-Pair Encoding (Sennrich et al., 2016) tokens. The weight of unsupervised loss λ is set to 1.0 in all our experiments, which is an empirical choice from previous studies (Sohn et al., 2020). The batch size is 8 for the supervised objective and 56 for the unsupervised objective, such that the model can leverage more unlabeled data for training. The threshold σ for the style strength filter is set to 0.8 and the threshold ratio ϕ is set to 0.4 for both the content preservation and fluency filters. We tested σ in the discrete range between 0.5 and 0.9 and for

ϕ, we searched over the values between 0.1 and 0.8. Although the chosen values of σ and ϕ are not necessarily the best for all the datasets, we fix them in later experiments for their reasonable results.

A.2 Training Details

We train two binary style classifiers on each domain of GYAFC. The training data are the formal and informal sentences in the original training sets of the E&M and F&R domain. The classifiers are validated on the formal sentences in the original validation set. The classifier for E&M could achieve 95.69% accuracy on the validation set, while the classifier for F&R achieved 94.70%. We adopt a 4-gram Kneser-Ney language model to compute perplexity scores for the fluency data filter. During semi-supervised training, we first pretrain the model solely on the supervised data for 2000 steps to achieve a good initialization of the model parameters. Then, we jointly train the supervised and consistency losses simultaneously. The model checkpoint is validated with an interval of 1000 steps and selected based on the best BLEU score on the validation set. Early stopping is also adopted with patience 10. We employ beam search with beam width 5 for the model's generations and pseudo-target prediction[5]. All our experiments are conducted on NVIDIA A100 (40GB) GPUs.

A.3 Details of Unlabeled Data Collection

We collected 200k from each of the E&M and F&R domains of Yahoo Answers L6 corpus. The collection procedure is as follows. (1) We chose the passages labeled "<bestanswer>" in the corpus and tokenized them into separate sentences. (2) We filtered out sentences with formality scores larger than 0.5 (i.e. judged as formal) predicted by the style classifier we built for model evaluation. (3) We built an N-gram language model by training on the informal sentences in the original training data of GYAFC, and used it to generate perplexity scores for these sentences. We kept 200k sentences with lowest perplexity scores, such that we obtained a collection of the most informal sentences in the corpus. We only observed one overlapping sentence with the test set of each domain, which we considered negligible and kept in the data.

[5]The pseudo target can also be obtained by sampling methods.

A.4 Details of Data Perturbation

All our data perturbation methods are implemented based on the nlpaug[6] library. We set the ratio of perturbed words in a sentence to 0.1 for all word-level perturbation methods and deduced that increasing the ratio could often result in lower results, as that will enhance the difference between the original and perturbed sentences, which is consistent with our conclusion in Section 4.2. We present examples of all data perturbation methods in Table 9.

We also attempted mixing different perturbations with *spell*, but did not obtain better results than single *spell*. This can also be attributed to the conclusion that simple perturbations are even better.

B Formal Description of the Algorithm

Here, we provide a formal algorithmic description of our consistency training framework in Algorithm 1 and assume that we adopt content preservation (BLEU) data filtering or fluency (perplexity) data filtering in this algorithm to include the formal description of our dynamic threshold strategy. We omit the case when we adopt style strength filtering because it does not use the dynamic threshold and is more straightforward to understand.

C Details of Dynamic Threshold Selection

Here, we provide more details of the dynamic threshold strategy for the content preservation and fluency filters. In practice, we do not filter any pseudo data in the initial warm-up steps of consistency training, to initialize the score list. Furthermore, after iterating an epoch of the unsupervised data, we keep the current threshold fixed and do not update the score list any more. The score list is implemented as a skiplist to enable $O(\log N)$ insertion into an ordered list. The overall time complexity of the data filtering is $O(\log 1 + \log 2 + \cdots + \log N) = O(\log N!) = O(N \log N)$, where N is the number of unlabeled data.

D Details of Human Evaluation

We describe the rating criteria in the human evaluation. We ask two well-educated annotators to rate the *formality*, *fluency*, and *meaning preservation* on a discrete scale from 0 to 2 for the model outputs, following (Zhang et al., 2020). During the annotation, we randomly shuffle the sentences

[6]https://github.com/makcedward/nlpaug

Original sentence	*Well first you have to get lots of hands on experience.*
Word deletion	*Well first you have to get lots of on experience.*
Word swapping	*Well first have you to get lots of hands on experience.*
Word masking	*Well first _ have to get lots of hands on experience.*
Word replacing with synonym	*Well first you have to begin lots of hands on experience.*
Back-translation	*well first you have to get lots of years on experience.*
TF-IDF based word replacing	*Well first you have walmartmusic get lots of hands on experience*
Spelling error injection	*Well first you have to get lots of hands or experience.*
Word replacing with abbreviations	*Well first u have to get lots of hands on experience.*
Word capitalization	*Well FIRST you have to get lots of hands on experience.*

Table 9: Examples of data perturbation methods. Different words compared to the original sentence are marked as red.

Algorithm 1 Training Procedure of our approach using dynamic threshold selection

Input: Parallel corpus $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}^M$, unlabeled corpus of source-side sentences $\mathcal{U}_S = \{\mathbf{u}\}^N$, initialized model parameters θ; perturbation function $c(\cdot)$, supervised batch size B, unsupervised batch size μB, weight factor λ, filter type $ft \in$ {BLEU, perplexity}, a data filter score function f, an decreasing-ordered score list L, a function $len(\cdot)$ that returns the length of a list.

▷ Warm-up training

Initialize θ with pretrained T5.
Finetune θ on $\mathcal{D}$ via Equation (1).

▷ Semi-supervised training

repeat
  Sample a batch $\mathcal{B}_D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{B}$ from $\mathcal{D}$.
  Sample a batch $\mathcal{B}_U = \{\mathbf{u}_i\}_{i=1}^{\mu B}$ from $\mathcal{U}_S$.
  Obtain $\mathcal{B}_U^{'} = \{\tilde{\mathbf{u}}_i | \tilde{\mathbf{u}}_i = c(\mathbf{u}_i)\}_{i=1}^{\mu B}$.
  Generate pseudo targets $\mathcal{B}_Y = \{\hat{\mathbf{y}}_i | \hat{\mathbf{y}}_i = \text{argmax} P(\mathbf{y}|\mathbf{u}_i; \theta)\}_{i=1}^{\mu B}$.
  Compute a batch of data filter scores $L_B = \{b_i | b_i = f(\mathbf{u}_i, \hat{\mathbf{y}}_i)\}_{i=1}^{\mu B}$
  Insert L_B into L while maintaining the decreasing order of L.
  Obtain $s = L[\sigma \times len(L)]$ as the threshold.
  if $ft =$ BLEU **then**
    Obtain a filtered pseudo-parallel batch $\mathcal{B}_f = \{(\tilde{\mathbf{u}}_i, \hat{\mathbf{y}}_i) | b_i > s, i = 1, , \ldots, \mu B\}$
  else if $ft =$ perplexity **then**
    Obtain a filtered pseudo-parallel batch $\mathcal{B}_f = \{(\tilde{\mathbf{u}}_i, \hat{\mathbf{y}}_i) | b_i < s, i = 1, , \ldots, \mu B\}$
  end if
  Compute consistency loss $\mathcal{L}_{unsup} = \mathbb{E}_{(\tilde{\mathbf{u}}, \hat{\mathbf{y}}) \sim \mathcal{B}_f}[-\log P(\hat{\mathbf{y}}|\tilde{\mathbf{u}}; \theta)]$.
  Compute supervised loss $\mathcal{L}_{sup} = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{B}_D}[-\log P(\mathbf{y}|\mathbf{x}; \theta)]$.
  Optimize $\mathcal{L} = \mathcal{L}_{sup} + \lambda \mathcal{L}_{unsup}$ and update θ.
until CONVERGE

from the three models and make the model names invisible to annotators.

Formality The annotator are asked to rate the formality change level given a source informal sentence and the generated output sentence, regardless of the fluency and meaning preservation. If the output sentence improves the formality of the source sentence significantly, the score will be 2 points. If the output sentence improves the formality but still keeps some informal expressions, or the improvement is minimal, it will be rated 1 point. If there is no improvement on the formality, it will be rated 0 points.

Fluency The fluency is rated 2 points if the output sentence is meaningful and has no grammatical error. If the target sentence is meaningful but contains some minor grammatical errors, it will be rated 1 point. If the sentence is incoherent, it will be rated 0 points.

Meaning Preservation Given a source sentence and a corresponding output sentence, the raters are asked to ascertain how much information is preserved in the output sentence compared to the input sentence. If the two sentences are exactly equivalent, the output obtains 2 points. If they are mostly equivalent but different in some trivial details, the output will receive 1 point. If the output omits important details that alter the meaning of the input sentence, it is rated 0 points.