ReMask: A Robust Information-Masking Approach for Domain Counterfactual Generation

Pengfei Hong[1*], **Rishabh Bhardwaj**[1*], **Navonil Majumdar**[1], **Somak Aditya**[2], **Soujanya Poria**[1]

[1] ISTD, Singapore University of Technology and Design, [2] Department of CSE, IIT Kharagpur
{pengfei_hong, navonil_majumder}@sutd.edu.sg
rishabh_bhardwaj@mymail.sutd.edu.sg
saditya@cse.iitkgp.ac.in, sporia@sutd.edu.sg

Abstract

Domain shift is a big challenge in NLP, thus, many approaches resort to learning domain-invariant features to mitigate the inference phase domain shift. Such methods, however, fail to leverage the domain-specific nuances relevant to the task at hand. To avoid such drawbacks, domain counterfactual generation aims to transform a text from the source domain to a given target domain. However, due to the limited availability of data, such frequency-based methods often miss and lead to some valid and spurious domain-token associations. Hence, we employ a three-step domain obfuscation approach that involves frequency and attention norm-based masking, to mask domain-specific cues, and unmasking to regain the domain generic context. Our experiments empirically show that the counterfactual samples sourced from our masked text lead to improved domain transfer on 10 out of 12 domain sentiment classification settings, with an average of 2% accuracy improvement over the state-of-the-art for unsupervised domain adaptation (UDA). Further, our model outperforms the state-of-the-art by achieving 1.4% average accuracy improvement in the adversarial domain adaptation (ADA) setting. Moreover, our model also shows its domain adaptation efficacy on a large multi-domain intent classification dataset where it attains state-of-the-art results. We release the codes publicly at https://github.com/declare-lab/remask.

1 Introduction

Despite significant advances in unsupervised representation learning, natural language processing (NLP) systems often strongly rely on expensive human-annotated datasets. These (labeled) datasets, however, are only available in specific domains. Systems trained on such datasets usually significantly under-perform on out-of-domain (OOD) samples during inference, due to strong reliance on the dataset-specific token- or feature-label correlations. These correlations do not often generalize beyond the domain of the training dataset. These correlations can even be spurious annotation artifacts. For example, in a sentiment-labeled dataset on restaurant reviews, the token *food* may frequently appear in samples with negative sentiment score, due to selection bias (Veitch et al., 2021). Such biases are particularly prevalent in low resource settings, where training instances are scarce (Nan et al., 2021).

In response, annotating samples from new domains may prove to be expensive and infeasible in the long term. To address these issues, many domain adaptations (DA) techniques have been proposed (Roark and Bacchiani, 2003; Daumé and Marcu, 2006; Ben-David et al., 2010; Jiang and Zhai, 2007; Rush et al., 2012; Schnabel and Schütze, 2014). Many of such techniques (Blitzer et al., 2007; Ziser and Reichart, 2016; Ganin et al., 2015; Ben-David et al., 2020) learn domain-invariant features that sidestep the pitfall of relying on domain-specific features when faced with OOD inputs. This obviously also hinders performance on in-domain samples where domain-specific features may be relevant. To address this issue, Calderon et al. (2022) recently proposed domain-counterfactual generation (DoCoGen) to transform given in-domain samples into out-domain samples.

DoCoGen masks the source-domain-specific n-grams in the input using a token-frequency-based approach. These masks are filled with target-domain-specific tokens using a conditional text generation language model that is fine-tuned with an unsupervised sentence reconstruction objective. However, the frequency-based masking approach does not account for the context to identify the source-domain-specific tokens. Moreover, this approach is limited by the statistics of an incomplete training set. These shortcomings may cause the ap-

[*] Equal Contribution

Findings of the Association for Computational Linguistics: ACL 2023, pages 3756–3769
July 9-14, 2023

proach to overlook many valid domain-token associations. For instance, in Fig. 1 the word *tearjerker* is not masked by DoCoGen, where this word may be associated with the DVD of a tragedy movie. Furthermore, such frequency-based masking fails to generalize well to out-of-vocabulary words.

Therefore, we propose a robust masking method that can achieve better retention of non-domain information, while removing domain-specific information. Our method includes three phases: frequency-based mask initialization (Calderon et al., 2022), over-the-top (OOT) masking, and unmasking. We first phase is simply to initialize the mask using the frequency-based strategy by Calderon et al. (2022). These masks are based on token-domain association consisting of two factors: the probability of the presence of a target token in a particular domain and the non-uniformity of this probability distribution. Tokens surpassing a certain association score, w.r.t. both source and target domain, are masked. In the second phase, we leverage the encoded knowledge in language models (LM) (Liu et al., 2019) to identify additional token-domain associations, improving recall at the risk of hurting precision. We found that a token with high attention-norm (Kobayashi et al., 2020), in a fine-tuned LM as domain classifier, could be strongly associated with the predicted domain. Such tokens with high enough attention norms are masked. Finally, we sequentially unmask the masked tokens, guided by the confidence score of a domain classifier. To minimize spurious token-domain associations from the prior two masking phases that hurt precision, the tokens that cause low domain confidence upon unmasking are kept unmasked.

The overall contributions of this work are: *i)* a robust masking strategy to mask more domain-specific tokens, as compared to the state of the art (SOTA) (Calderon et al., 2022), while mitigating spuriously masked tokens; *ii)* domain counterfactuals (D-CON) from our model outperform the SOTA on binary and multi-label classification tasks, under unsupervised domain adaptation (UDA) and adversarial domain adaptation (ADA) settings, under almost all domain shifts.

2 Methodology

Task Description. Let $X_{\mathcal{D}}$ denote a text sampled from domain $\mathcal{D}$. We define a counterfactual generator function $f_c^{\mathcal{D}\to\mathcal{D}'}(X_{\mathcal{D}})$ as a mapping from a text $X_{\mathcal{D}} \sim \mathcal{D}$ to a text $X_{\mathcal{D}'} \in \mathcal{D}'$, such that, all but domain $\mathcal{D}$-specific information is preserved in $X'_{\mathcal{D}}$. We denote the domain-agnostic information by X and the domain-specific text to be a specific fusion of X and $\mathcal{D}$. We define a fusion function $\mathcal{I}$, such that, $X_{\mathcal{D}} := \mathcal{I}(X; \mathcal{D})$. An inverse mapping $\mathcal{I}^{-1}$ makes it feasible to efficiently disentangle and extract the domain-generic information X. Therefore, the task of domain counterfactual generation aims to find a function that essentially disentangles the domain-generic information from $X_{\mathcal{D}}$ to obtain X, followed by mapping X to a domain-specific text $X_{\mathcal{D}'}$. Thus, an ideal $f_c^{\mathcal{D}\to\mathcal{D}'}$ can be considered to be functionally equivalent to $\mathcal{I}(\mathcal{I}^{-1}(X_{\mathcal{D}}), \mathcal{D}')$.

Overview. The proposed approach consists of two phases:

• *Domain Obfuscation*: This phase masks a text $\mathcal{X}_D$, such that, it becomes void of source-domain-specific information. To this end, we propose a three-step masking approach.

• *Domain Counterfactual Generation*: As Calderon et al. (2022), we feed the masked text to a T5-based encoder-decoder model f_c to generate a counterfactual in the target domain $\mathcal{D}'$.

2.1 Domain Obfuscation

We propose a novel three-step approach for domain corruption to mask more words carrying domain-specific cues and unmask words that are not specific to any domain. Masking words carrying domain-specific information helps the model learn to rely more on the input prompt, allowing us to have higher control over the generated text. Moreover, unmasking domain-generic words preserve the context information of text so as to keep the model-generated output an equivalent text to input but in the specified domain.

Step 1 (Base Masking). We begin with the frequency-based (heuristic) masks, proposed by Calderon et al. (2022). It assigns an affinity-based score to a word $w \in X$. The word (w)-domain($\mathcal{D}$) affinity is defined by

$$\rho(w, \mathcal{D}) = P(\mathcal{D} \mid w) \cdot \left(1 - \frac{H(D \mid w)}{\log N}\right),$$

where $D \in \{1, \ldots, N\}$ is a random variable representing a N domain classes. $H(D|w)$ denotes the entropy of random variable $D|w$. A low value of $H(D|w)$ signifies that the spread of word w is skewed towards fewer domains, and thus domain-specific. $P(\mathcal{D}|w)$ dictates the chances of domain

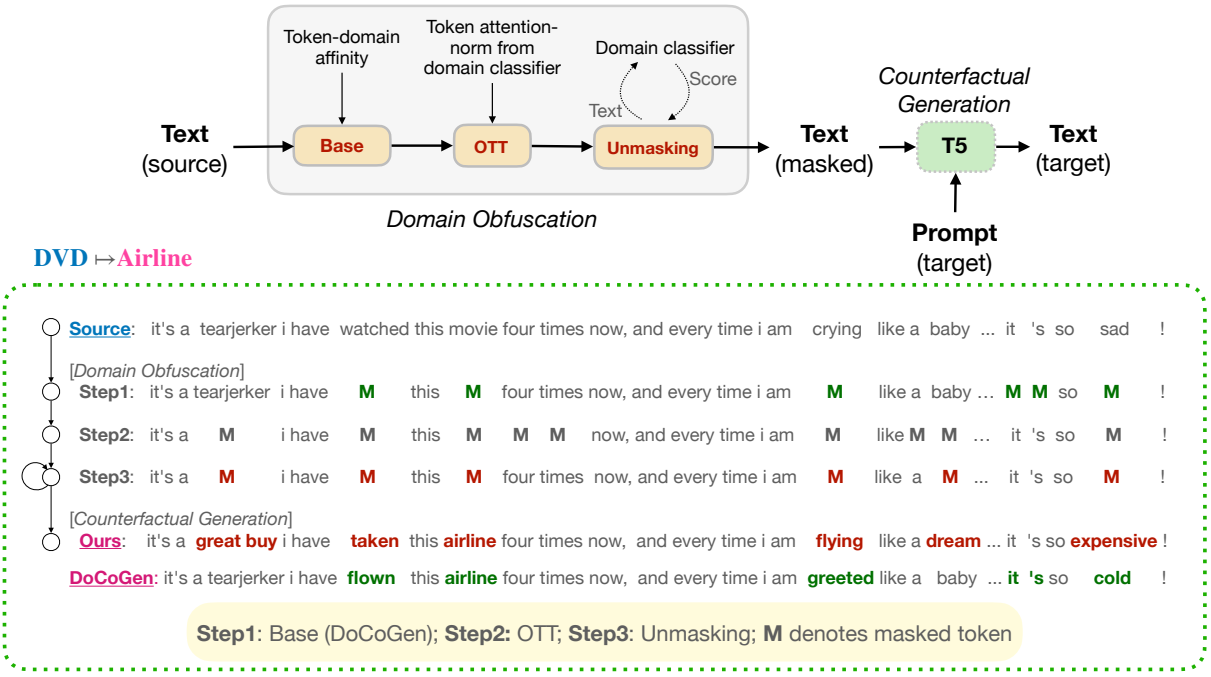

Figure 1: This diagram outlines the flow of our approach (top part) and illustrates it with an example (bottom part); the input **Text** (source) is sequentially fed through *Base*, *OTT*, and *Unmasking* steps to produce source-domain unaware masked text, which is passed to a T5-based generator to obtain domain counterfactual.

being $\mathcal{D}$ given the word w. Thus, a word is said to have a high affinity with domain $\mathcal{D}$ if $\rho(w, \mathcal{D})$ returns a high value. To perform $X_{\mathcal{D}} \rightarrow X'_{\mathcal{D}}$ counterfactual generation, we mask a word $w \in X$ if the word has a high affinity towards $\mathcal{D}$, relative to $\mathcal{D}'$. Thus, the domain transfer (masking) score can be defined by

$$\mathrm{m}_a\left(w, \mathcal{D}, \mathcal{D}'\right) := \rho(w, \mathcal{D}) - \rho\left(w, \mathcal{D}'\right).$$

The higher the value of $\mathrm{m}_a\left(w, \mathcal{D}, \mathcal{D}'\right)$, the harder it is to perform its $\mathcal{D} \rightarrow \mathcal{D}'$ transfer. A word is masked if its m_a-score is above a predefined threshold τ_1. Generalizing the approach to n-grams, we first mask all the unigram words. This is followed by identifying the bigrams that do not overlap with the masked unigram, and then trigrams that do not overlap with the masked tokens[1].

Step 2 (Over-The-Top Masking). After heuristic masking, which is word context agnostic, we perform attention-based over-the-top (OTT) masking. The OTT masking serves two purposes—1) Context awareness: based on their context information, it reconsiders the words for masking that failed m_a-scoring and remained unmasked; and 2) Induction: it generalizes the notion of a domain to a more comprehensive set of word vocabulary and their word co-occurrences in which the domain-specific corpus is prone to be deficient. The context-aware attribute masks generic words, that under a given context and style of text, become domain informative. The inductive masking attribute makes use of the underlying language model to exploit the learned word-word co-occurrences to reconsider the unmasked words. Thus, considering words missing in the domain-specific corpus as well as words whose frequencies in the corpus do not truly reflect their domain affinity. This results in improved recall in domain-specific token extraction, although at the risk of hurting precision.

To perform OTT masking, we train a domain classifier $f_d : \mathcal{D} \rightarrow D$ that learns to classify an $X_{\mathcal{D}} \sim \mathcal{D}$ to a domain label $d \in D$. For finding OTT words to mask, we use an attention-based architecture as domain classifier f_d. Norm-based attention analysis has been seen to efficiently capture the importance of a word to the model's prediction (Kobayashi et al., 2020). Following this observation, we define the attention-based domain affinity of a word as

$$\mathrm{m}_b\left(w; l\right) := ||\alpha_w^l \boldsymbol{v}_w^l||,$$

where $||\cdot||$ computes Euclidean norm. In layer l operations, let $\mathbf{W}_V^l$, $\boldsymbol{b}_V^l$ denote the value matrix

[1]Threshold τ_1 is a hyperparameter, best found to be 0.08

and biases, $\mathbf{W}_O^l$ be the output projection. For a word w, let $\boldsymbol{y}_w^l$ be its representation at the output of layer l, we define $\boldsymbol{v}_w^l = (\boldsymbol{y}_w^{l-1}\mathbf{W}_V^l + \boldsymbol{b}_V^l)\mathbf{W}_O^l$, α_w^l specifies how much the classification token[2] in layer l attends to w. We mask all the words whose m_b score is higher than a threshold denoted by τ_2. τ_2 is set as a hyperparameter based on each $X_\mathcal{D}$, this will be explained further in step 4.

Step 3 (Unmasking). One of the aims of domain-counterfactual text generation is to preserve everything —- including the task label — except domain information. Therefore to make the minimal edits to its original text (Calderon et al., 2022), we reduce the number of edits to by restoring the masked tokens from prior steps. Meanwhile, we posit that domain information masked by OTT (Step 2) is not unmasked in Step 3. The latter is controlled by thresholding the domain classification score of the masked sentence produced in this step 2. Analogous to counterfactually-augmented data (CAD) generation (Kaushik et al., 2020), which is done by minimally intervening on examples to change their ground truth label. In our case, we only want to use a minimal number of masks to remove the amount of domain-specific terms in the text. Specifically, we use the domain score given by the domain classifier to measure the amount of domain information contained in the example and make sure it is below a threshold [3].

Additionally, Step 1 and Step 2 are prone to excessive masking from spurious domain-token correlations, resulting in the improved recall at the risk of poor precision. To account for these factors, we aim to unmask tokens that do not provide domain-specific cues, hence, restoring more contextual information; this is inspired by Meng et al. (2022), where *causal tracing* is used to determine the association between different factors and the output in a large language model. Given input text $X_\mathcal{D}$, following Step 1 and Step 2, we obtain a masked text $\tilde{X}_\mathcal{D}^K$, where $K = \{k_1, ..., k_n\}$ represents n positions of the masked tokens. We define a subset of indices $U = \{k_i, ..., k_j\} \subseteq K$ containing the positions of the words to be unmasked. Thus, $\tilde{X}^{K-U}$ represents the rectified masked output.

First, we independently intervene on the masked text $\tilde{X}_\mathcal{D}$ by unmasking a token at position i and perform domain classification using f_d, to obtain the change in domain-label probability of D. We also define

$$\mathrm{m}_u(w_{k_i}) := f_d^\mathcal{D}(\tilde{X}_\mathcal{D}^{K-\{k_i\}}) - f_d^\mathcal{D}(\tilde{X}_\mathcal{D}),$$

where $f_d^\mathcal{D}(\cdot)$ returns a probability score of the origin domain $\mathcal{D}$. Per iteration, we sequentially restore the tokens in K, in the ascending order of $\mathrm{m}_u(w_{k_i})$. We stop the unmasking iteration when the condition $f_d^\mathcal{D}(\tilde{X}_\mathcal{D}^{K-U}) < \tau_3$ is violated. Then we get the unmasked example. We posit that the domain specificity is highly correlated with $\mathrm{m}_u(w_{k_i})$. Therefore, we find taking the greedy approach to unmasking from the most domain-descriptive tokens robust in finding the optimal U.

2.2 Domain Counterfactual Generation

Training. We do not have access to parallel domain counterfactuals. Thus, following Calderon et al. (2022), we train a T5-based encoder-decoder model M in an unsupervised manner. We essentially train the model to reconstruct the original text X in the source domain from its masked form $\tilde{X}$, which is presumably purged of source-domain-specific cues. To have control on the domain of the generated text, we follow DoCoGen's approach of prepending a soft prompt v, indicating the domain, to the masked text in the input. The soft prompt is initialized using the embedding of the word[4] representative of the target domain $\mathcal{D}$ and trained together with the text reconstruction objective. During training, the model learns to generate the original example, given the domain obfuscated text and the soft prompt: $M(\tilde{X}, v) \rightarrow X$.

Inference. Given $(x, \mathcal{D}, \mathcal{D}')$, we first feed through the x through our domain obfuscation process to get $\tilde{x}$ and select the soft prompt v' to represent $\mathcal{D}'$. We feed $(\tilde{x}, v')$ to the model to generate the domain counterfactual x'. We used beam search with a beam size of four for decoding x'.

3 Intrinsic Evaluation

Conditional generation is difficult to automatically evaluate. Therefore, we asked three Ph.D. students trained in natural language processing to manually evaluate the dataset considering the following evaluation measures: (1) Domain Relevance

[2] Classification token for RoBERTa is <s>

[3] In our initial experiments, we found the threshold value 0.4 works best. we also tried thresholding based on misclassification, in our observations, it tends to unmask a lot of context-specific words.

[4] Following DoCoGen, we choose the most common word in that domain as the representative of that domain, we list the words we use in the appendix

Original from **[dvd]**: Pippin dvd, I loved Ben Vereen in the show. Wanted the music for my ipod, too. Very satisfying!
Masking [DoCoGen] to **[book]**: Pipin <m>, I loved Ben Vereen in the <m>. Wanted <m> <m> for my ipod too. Very satisfying! Masking [Ours] to **[book]**: Pipin <m>, I loved <m> <m> in the <m>. Wanted the for my <m> too. Very satisfying!
Original from **[electronics]**: sony rm - ax4000 this is a poorly designed remote. i have six devices connected to my hd television set. the software depends on the user assigning positions to the various inputs to the tv and, in my case, routinely activated the wrong device when i used the remote. i purchased a logistics harmony remote and it works perfectly.
Masking [DoCoGen] to **[kitchen]**: <m> - ax4000 this is a poorly <m> . i have six <m> to my <m> set . the <m> depends on the <m> assigning positions to the various <m> to <m> , in <m> routinely <m> wrong <m> when <m> the <m> . <m> logistics <m> and <m> . Masking [Ours] to **[kitchen]**: <m> - <m> this is a poorly designed <m> . i have six <m> to my <m> set . the <m> depends on the <m> assigning <m> to the various <m> , in my case , routinely <m> the wrong <m> when i <m> . i purchased a <m> and it works perfectly.

Table 1: Domain Obfuscation (Masking).

Model	↑ D.REL	↑ L.PRES	↑ ACCPT	↓ WER
DoCoGen	85	87.5	4.21	1.05
ReMask	92.5	95.0	4.33	1.0
Original Reviews	100.0	100.0	4.94	0

Table 2: Human intrinsic evaluation. Up arrows (↑) represent metrics where higher scores are better, and down arrows (↓) represent the opposite.

(D.REL) - whether the topic of the generated text is related to the target domain; (2) Label Preservation (L.PRES) - what is the label of the generated text and if the original label preserved; (3) Linguistic Acceptability (ACCPT) - how logical and grammatical the example is (on a 1-5 scale); and (4) Word Error Rate (WER) - what is the minimum number of word substitutions, deletions, and insertions required to make the example logical and grammatical. The test is conducted on 20 reviews, uniformly distributed among four domains (A, D, E, K) and 60 generated domain counterfactuals using DoCoGen and ReMask.

Table 2 shows that our masking achieves better scores than its baseline DoCoGen method, especially in preserving the original example label, it achieves a score of 95%, surpassing its heuristic counterpart. We suspect ReMask keeps more information for the generation model to infer the label thus preserving label-invariance. We analyse the reason for this further in Section 6

4 Experimental Settings

4.1 Tasks and Settings

We focused on two low-resource scenarios: unsupervised domain adaptation (UDA) and any domain adaptation (ADA). UDA assumes the availability of large unlabeled data from a source domain and target domain, as well as access to the limited number of labeled examples in the source domain. We follow Calderon et al. (2022) and choose to sample 100 labeled data in the source domain for both of the tasks mentioned below. An even more challenging and potentially more realistic setting is ADA, which assumes no access to both labeled data and unlabeled data from the target domain in training time. In other words, the model will not have any information about the target domain distribution. Following a large body of DA work, we benchmark the proposed ReMask on cross-domain sentiment classification task and cross-domain multi-label intent classification task. For brevity, we refer readers to Calderon et al. (2022) for a detailed description of each dataset.

Sentiment Classification In this task, we combine 3 datasets together to form a dataset of 6 domains. The dataset includes 4 domains - Books(B), DVDs(D), Electronic items(E), and Kitchen Appliances(K) from product review multi-domain dataset (Blitzer et al., 2007); the challenging airline review dataset (A) (air); and the restaurant (R) domain from Yelp dataset challenge (Wei and Zou, 2019). We benchmark the sentiment classification task on UDA and ADA settings. In UDA, we focus on four of the six domains: A, D, E, and K. This will result in 12 (3×4) cross-domain pairs. In this setting, the model can access all of the unlabeled source domain and target domain data. whereas For ADA, where an unlabeled dataset from the target domain is not within reach, our experiment uses A, D, E, and K as source domains; B, and R as target domains, thus resulting in a total of 8 (4×2) domain pairs for ADA. Further, To facilitate the comparison with previous work, we focused on low resource settings by randomly sampling 100 labeled examples each from the four domains to as labeled source dataset. The reported scores are averaged across 5 training and development sets using different seeds.

Multi-Label Intent Prediction. The second task is to predict the potential intents of the utterances arising from information-seeking dialogs, as

there could be multiple intents originating from the same utterances, it is treated as a multi-label classification problem. We choose the MANtIS dataset (Penha et al., 2019), inside each utterance could have eight potential intent labels. Following Calderon et al. (2022), we only consider 5 most common labels: Further Detail (FD), potential answer (PA), Information Request (IR), Greetings and Gratitude (GG), Original Question (OQ).
The MANtIS dataset is a multi-domain dataset with 14 domains. We experiment with the UDA setting. To ease comparison with DoCoGen, we choose the first six domains: Apple (AP), DBA (DB), Electronics (EL), Physics(PH), Statistics (ST), askubuntu (UB) with available unlabeled data to form 30 (5×6) cross-domain pairs for UDA.

5 Models and Baselines

We tested on three types of models [5]: (a) baseline models (b) variants for each step in our masking approach c) an upper-bound model to approximate the best performance using domain counterfactuals (D-CON) on the downstream tasks. Unless otherwise stated, all the domain classifiers and sentiment classifiers use the same model, based on a pretrained Roberta-base model, and all generation models are based on a pretrained `T5-base`.

Baseline DA Models. We experiment with four baselines: (1) No-Domain-Adaptation (NoDA), the model only trained on available training data from the source domain; (2) Random-masking Random-Reconstruction (RMRR) (Ng et al., 2020), randomly masks token from the input example and then fills the masks by masked language modeling head. 3) PERL (Ben-David et al., 2020), a SOTA for the UDA setup. 4) DoCoGen: Use the same model and training strategy as we do, but only use the first step of our masking procedure.

Ablations. We consider three types of ablations in our masking procedure. First, we show ablations for step 2 in our procedure. (5) 2-Attention-Score-Masking (2-ASM) where we use attention score rather than attention norm to perform OOT masking step, (6) 3WO where we show the performance of unmasking by following word order in the sentence. And finally, we show (7) NoInit, where we remove the heuristic masking step, and only use OOT masking with unmasking. (8) No-Unmask, where we only have step 1,2 and removed step 3 unmasking.

Upper-Bound. To approximate the upper-bound for D-CON augmentation, we use *Oracle-Matching* (Oracle) which can access to target domain labeled data (Calderon et al., 2022). Given an example from a source domain, Oracle looks for the most similar example with the same label in the target domain as training data.

6 Results and Discussions

Tables 3 and 4 present sentiment classification accuracy results for the 12 UDA and 8 ADA setups respectively. Table 5 presents the average intent prediction F1 scores for each source domain, taken across 5 target domains for the UDA setup.

Results Comparison. For sentiment classification, our method ReMask, outperforms all baseline models in 10 of 12 UDA setups and in 6 of 8 ADA setups, gaining an average of 2% and 1.4% over the baseline masking approach in UDA and ADA settings, respectively. Furthermore, in two ADA setups on sentiment classification, our model outperforms the Oracle-Gen scores which suggests our model can effectively remove domain-specific information in these domain pairs. For intent prediction [6], our model outperforms four out of six setups, reaching an average gain of 0.8% across domains. Moreover, NoInit result demonstrates that our model performance degrades if there is no initial mask provided by step 1. One key observation on the masking process is that Step 2 of our method fails to identify some domain-specific tokens in the presence of too many such tokens in the text. We suspect that the thresholding attention norm is not robust enough when the attention norm might be thinly spread among many domain-specific tokens. On the other hand, ReMask outperforms DoCoGen by 0.8% on the MANtIS dataset under the UDA setup. This further confirms the efficacy of our method. We found that our ReMask have less performance improvement for domains where the domain specific words are distinctive like UB and have higher performance boost in domains that have more complex text, e.g. EL and PH.

Discussion on ReMask's ability to retain contextual information and remove domain-specific

[5] Experiment 3 is not applicable to the ADA setting due to lack of labeled data.

[6] lack of working code implementation and specifications to reproduce the reported results of `DoCoGen`

Source Domain → Target Domain ($\mathcal{D} \rightarrow \mathcal{D}'$)													
Model	**A→D**	**A→E**	**A→K**	**D→A**	**D→E**	**D→K**	**E→A**	**E→D**	**E→K**	**K→A**	**K→D**	**K→E**	**AVG**
NoDA	69.4	78.6	78.2	72.3	80.2	82.4	81.0	79.8	87.6	72.5	78.6	85.4	78.8
RM-RR	69.5	80.1	80.0	72.3	81.0	83.8	79.6	79.5	88.4	70.6	79.1	84.5	79.0
PERL	72.9	81.1	83.6	81.5	83.0	86.9	81.1	81.7	88.5	77.9	78.2	86.1	81.9
DoCoGen	70.6	79.7	79.8	75.8	82.8	84.4	83.0	82.0	89.3	81.2	82.2	87.3	81.5
DoCoGen-Rob	62.7	73.9	74.9	76.9	85.7	87.7	82.7	82.1	90.3	81.9	83.0	88.9	80.9
2-ASM	68.7	81.8	79.5	81.0	88.3	88.9	82.0	84.4	91.4	80.7	80.9	88.1	82.9
3WO	65.4	78.4	81.5	80.3	81.0	80.0	85.4	75.6	85.2	84.2	80.0	82.1	79.9
NoInit	62.7	73.9	74.9	76.9	85.7	87.7	82.7	82.1	90.3	81.9	82.95	88.9	80.9
NoUnmask	69.1	77.2	71.1	78.5	79.6	78.9	76.4	80.0	83.4	66.4	74.1	82.0	76.4
NoOTT	62.7	73.9	74.9	76.9	85.7	82.7	82.1	90.3	81.9	83.0	88.9	80.9	80.8
OUR	**74.4**	**85.3**	78.5	**78.2**	**88.5**	**88.9**	81.0	**82.9**	**89.7**	**81.5**	**83.4**	**89.7**	**83.5**
Oracle-Gen	83.8	88.4	88.9	83.6	89.3	90.0	84.9	84.6	90.7	84.1	82.2	89.0	86.6

Table 3: Unsupervised Domain Adaptation results for sentiment classification in the (Blitzer et al., 2006) dataset. DoCoGen-Rob uses a RoBERTa-base task classifier instead of T5 for a fair comparison to our model.

Source →	**A**		**D**		**E**		**K**		***AVG***
Target →	**B**	**R**	**B**	**R**	**B**	**R**	**B**	**R**	
NoDA	69.1	76.5	82.3	82.8	81.5	84.5	82.4	85.2	80.5
RM-RR	69.4	78.4	83.8	83.5	81.9	85.6	83.7	85.4	81.5
DoCoGen	70.9	78.1	84.4	82.9	83.9	86.0	84.5	85.7	82.1
DoCoGen-Rob	70.4	82.1	79.5	82.4	82.9	86.6	85.5	86.7	82.0
2-ASM	65.7	76.3	87.1	82.3	80.8	84.9	85.0	82.3	80.7
NoInit	71.4	82.4	87.3	82.2	84.0	84.3	81.5	85.2	82.3
NoUnmask	62.4	74.9	80.4	75.3	74.2	74.0	80.5	79.4	75.1
Our	72.9	83.4	91.6	82.4	84.0	85.5	82.5	86.3	83.5
Oracle	84.4	85.2	91.6	86.1	86.0	86.5	85.3	86.5	86.4

Table 4: Adversarial Domain Adaptation (ADA) results on the (Blitzer et al., 2006) dataset.

Model	**AP**	**DB**	**EL**	**PH**	**ST**	**UB**	***AVG***
NoDA	71.1	67.1	72.0	47.3	66.0	72.8	66.1
DoCoGen	69.8	73.9	70.7	62.0	67.8	73.4	70.1
ReMask	70.7	73.2	72.7	60.3	68.1	73.4	70.9
Oracle-Gen	77.1	75.0	76.2	73.6	70.1	72.3	74.1

Table 5: UDA results on the MANtIS dataset (2019).

Model / #Train Samples	**400**	**1k**	**10k**
DoCoGen	40.6	70.4	86.6
ReMask	15.4	44.1	85.2

Table 6: Domain classification accuracy on the masked text between ReMask vs DoCoGen masking.

information. To test our proposed method's effectiveness in removing domain-specific information, we train a domain classifier (based on the Roberta-base model) to predict the domain from the masked text. Table 6 shows that the accuracy of the domain classifier trained on text masked by ReMask is consistently lower than DoCoGen, which proves the efficiency of our model ability in the obfuscation of the domain. The difference shrinks as the number of training examples increases. This may suggest that the model is still able to exploit the spurious correlation between linguistic features and the domain of the dataset.

The Need of Step 2 and Step 3. As presented in Table 3, ReMask without Step 3 (NoUnmask) performs very poorly, worse than DoCoGen. This might give us the impression that Step 2 is unnecessary. However, we hypothesize that Step 1 alone is not enough to correctly distinguish all the domain-indicative tokens — as Step 1 is not context-sensitive. Step 2 addresses this issue which essentially aims at increasing the recall of the token-level masking. Although Step 2 can increase recall

Domain	A			D			E			K		
	Step1	+Step2	+Step3	Step1	+Step2	+Step3	Step1	+Step2	+Step3	Step1	+Step2	+Step3
A	15.2	27.8	16.3	37.9	48.16	44.5	37.3	54.2	30.9	38.0	56.1	33.3
D	25.0	34.1	31.3	16.5	27.6	24.0	24.0	44.5	23.2	23.9	45.2	26.1
E	27.8	33.4	26.5	26.7	30.1	24.5	15.7	30.2	16.4	19.7	22.0	19.6
K	30.2	40.2	38.7	28.7	45.8	35.6	21.1	33.5	21.4	15.7	27.0	22.8

Table 7: The average number of masks in each step.

of toke-level masking, it has no control over the precision and hence can produce imperfect masks that are domain neutral or key to preserving contextual cues. One should also note that counterfactual generators should try to generate counterfactuals with minimal edits. So, we try to spot and remove these imperfect masks by a greedy unmasking strategy in Step 3. As such, Step 3 aims to produce masks without losing recall in Step 2 (OTT), but to improve the precision of Step 2-produced masks significantly. We empirically verify the functioning of these steps in Table 7. We can see that the number of masks after Step 2 increases significantly, thus risking low precision and high recall (comparable to information retrieval problems). Step 3 unmasks such imperfect token-level masks resulting in fewer token-level masks than Step 2 in Table 7. We also show qualitative examples in Table 8 to depict the usefulness of Step 2 and Step 3, where Step 3 unmasks critical domain neutral and contextual cue-bearing tokens as well as correctly retains important domain-dependant masks. This, as a result, maintains the recall of step 2 but increases its precision. Although we do not have a direct measurement of this precision and recall hypothesis due to the lack of ground truth labels of these masks — we indirectly prove so using our experiments in Table 3. The number of masks after Step 3 lies between Step 1 and Step 2 indicating Step 1 may not capture all the relevant masks and Step 2 might be excessively masking. If we further link this observation with Table 3, we can conclude that Step 3 is required to improve performance over Step 1 (DoCoGen).

Source domain: dvd; destination domain: electronics
Original Text: Pippin dvd, I loved Ben Vereen in the show. Wanted the music for my ipod, too. Very satisfying!
First Step Maksing: Pipin <m>, I loved Ben Vereen in the <m>. Wanted the <m> for my ipod, too. Very satisfying!
Second Step OTT Masking: <m> <m>, I loved <m> in the <m>. Wanted the <m> for my ipod, too. Very <m>!
Third Step Unmasking: Pipin <m>, I loved <m> in <m>. Wanted the <m> for my ipod too. Very satisfying!
Generation: Pipin is great, I loved green color in the earphone. Wanted the earphones for my ipod too. Very satisfying!

Source domain: airline, destination domain: kitchen
Original Text: 40 am departure time became 4. 30 am due to a crew issue. the plane was an older 737. the empower jacks at our seats were dead and there was grime on the hard part of the seat. there was no special meal service for business class - everybody got a small ham and cheese sandwich. the front toilet quickly got foul.
First Step Heuristic Masking:
<m> time became 4 <m> to a <m> issue . the <m> . the empower jacks at our <m> dead and <m> grime on the hard part of the <m> . <m> special <m> for <m> - everybody got a small <m> and <m> . the <m> quickly got foul.
Second Step OTT Masking:
<m> <m> became 4 <m> to a <m> <m>. the <m> <m> <m> . the <m> at our <m> <m> and <m> grime on the hard part of the <m> . <m> special <m> for <m> - <m> got a <m> <m> and <m> . the <m> quickly got <m>.
Third Step Unmasking:
<m> <m> became 4 <m> to a <m> issue. the <m> was <m> . the <m> at our dead and <m> grime on the hard part of the <m> . <m> special <m> for <m> - <m> got a <m> and <m> . the <m> quickly got <m>.

Table 8: A few examples of the masks produced by different steps of our approach; the masks highlighted with <m> are masks added in Step 2, but removed(unmasked) in Step 3, whereas <m> remains masked after Step 3; consecutive masks are at times merged into one mask for brevity.

7 Related Work

Our work takes inspiration from several established lines of research, namely Domain Adaptation, Counterfactual Data augmentation, and counterfactual text generation.

Domain Adaptation. Domain Adaptation (DA; (Farahani et al., 2021)) deviates from the assumption that test data comes from the same distribution as training and is aimed at improving the performance of models in target domains (possibly) with a different distribution than the training source domain. In NLP, various DA setups are considered Approaches in Unsupervised Domain Adaptation (Blitzer et al., 2006) assumes the availability of unlabeled data from both source and target domains, as well as the existence of labeled data in the source domain. In adversarial domain adaptation (Wang et al., 2019), the assumptions are the same as UDA, but adversarial learning is employed to learn domain-invariant representations. In contrast, and domain adaptation (ADA; (Ben-David et al., 2022)) approaches assumes no knowledge of the target domains at training time. We consider UDA and Adversarial DA settings in our work.

Counterfactual Data Augmentation. Data augmentation, in general, is an important sub-field of NLP that aims to mitigate problems introduced by

the low-resource availability of text data. It aims to increase the number of examples for training without any explicit efforts to collect new samples. There have been rule-based approaches that work by modifying the underlying text by using some pre-defined heuristics (Wei and Zou, 2019). Beyond significant heuristic efforts, there have been model-based approaches to modify the words or generate prior distribution-based completely new samples (Kobayashi, 2018). Counterfactual Data Augmentation is a more sophisticated approach that performs a minimal text intervention (such as a specific concept), thus constructing an example with a modified label (Kaushik et al., 2020).

(Counterfactual) Controlled Text Generation. In conditioned or controllable text generation (Prabhumoye et al., 2020), the task is to generate text that satisfies certain pre-specified conditions (such as topic, sentiment or domain). Its useful for a vast range of applications including data augmentation and domain adaptation. Previous approaches involved finetuning the LM outputs using re-inforcement learning (Ziegler et al., 2019), training Generative Adversarial Networks (Yu et al., 2017), or training conditional generative models. Recent approaches adapted causal text modeling (Feder et al., 2022) and uses counterfactual editing of text towards controlled text generation (Madaan et al., 2021; Wu et al., 2021). Most of these models experiment with shorter text and properties that are easier to *ground* in the text (more specifically replacing one or two spans suffice).

8 Conclusion

This work empirically shows that the addition of attention norm-based masking leads to additional domain-token associations missed by the SOTA masking strategy. Furthermore, our iterative unmasking approach leads to the removal of spurious domain-token associations, resulting in improved domain obfuscation. These two innovations collectively allow better domain transfer on sentiment and intent classifications tasks.

Limitations

We discuss the limitations of our work:

- While the three-step masking is shown to be beneficial, masking (base+OTT) followed by unmasking may introduce several redundant computations as a token masked in the first two steps might get unmasked in step 3.
- When compared against the DoCoGen baseline (Calderon et al., 2022), iterative masking steps increase the time complexity of the domain obfuscation which leads to masking latency. Moreover, as the domain classifier is a critical part of the domain obfuscation, the approach has extra memory footprints.
- The proposed masking approach introduces two extra hyperparameters τ_2 and τ_3 on top of the hyperparameters introduced by DoCoGen. While we identify them as a fixed scalar value working for all kinds of input, we posit that one can propose dynamic input or source domain adaptive thresholding. Currently, we classify it as a limitation of the proposed work.

Ethics Statement

For intrinsic evaluation, we engage three Ph.D. students who are fairly compensated. This qualitative evaluation project passed ethics review of our IRB as it does not contain any confidential data.

Acknowledgement

This research is supported by the Ministry of Education, Singapore, under its AcRF Tier-2 grant (Project no. T2MOE2008, and Grantor reference no. MOE-T2EP20220-0017), and A*STAR under its RIE 2020 AME programmatic grant (project reference no. RGAST2003. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

References

Skytrax user reviews dataset. https://github.com/quankiquanki/skytrax-reviews-dataset. Accessed: 2010-09-30.

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2022. PADA: example-based prompt learning for on-the-fly adaptation to unseen domains. *Trans. Assoc. Comput. Linguistics*, 10:414–433.

Eyal Ben-David, Carmel Rabinovitz, and Roi Reichart. 2020. Perl: Pivot-based domain adaptation for pre-trained deep contextualized embedding models. *Transactions of the Association for Computational Linguistics*, 8:504–521.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando C Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning*, 79:151–175.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. pages 440–447.

John Blitzer, Ryan T. McDonald, and Fernando C Pereira. 2006. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing*.

Nitay Calderon, Eyal Ben-David, Amir Feder, and Roi Reichart. 2022. Docogen: Domain counterfactual generation for low resource domain adaptation. *arXiv preprint arXiv:2202.12350*.

Hal Daumé and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *J. Artif. Intell. Res.*, 26:101–126.

Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. 2021. A brief review of domain adaptation. *Advances in data science and information engineering*, pages 877–894.

Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2022. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Trans. Assoc. Comput. Linguistics*, 10:1138–1158.

Yaroslav Ganin, E. Ustinova, Hana Ajakan, Pascal Germain, H. Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. 2015. Domain-adversarial training of neural networks. In *Journal of machine learning research*.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. *arXiv preprint arXiv:2004.10102*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. 2021. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13516–13524. AAAI Press.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35.

Guoshun Nan, Jiaqi Zeng, Rui Qiao, Zhijiang Guo, and Wei Lu. 2021. Uncovering main causalities for long-tailed information extraction.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness.

Gustavo Penha, Alex Bălan, and Claudia Hauff. 2019. Introducing mantis: a novel multi-domain information seeking dialogues dataset. *ArXiv*, abs/1912.04639.

Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1–14, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Brian Roark and Michiel Bacchiani. 2003. Supervised and unsupervised PCFG adaptation to novel domains. pages 205–212.

Alexander M. Rush, Roi Reichart, Michael Collins, and Amir Globerson. 2012. Improved parsing and pos tagging using inter-sentence consistency constraints.

Tobias Schnabel and Hinrich Schütze. 2014. FLORS: Fast and simple domain adaptation for part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 2.

Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. 2021. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *ArXiv*, abs/2106.00545.

Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang. 2019. Adversarial domain adaptation for machine reading comprehension. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2510–2520. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. pages 6382–6388.

Tongshuang Wu, Marco Túlio Ribeiro, Jeffrey Heer, and Daniel S. Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6707–6723. Association for Computational Linguistics.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593.

Yftah Ziser and Roi Reichart. 2016. Neural structural correspondence learning for domain adaptation. *ArXiv*, abs/1610.01588.

A Appendix

A.1 Hyperparameters and Setups

Data Preprocessing We follow previous work in data preprocessing and truncate each example to 96 tokens, using the HuggingFace T5-base tokenizer. The hyper-parameter was set to 96 due to computation reasons and since the median number of words in the labeled examples was 89. When an example is longer than 96 tokens, we keep the first 96 tokens. For example from the Airline domain, before truncating, we remove the first sentence since it mostly contains details about the flight (like "from JPK to LAX").

Heuristic Masking We estimate $P(\mathcal{D}|w)$ for unigrams, bi-grams and tri-grams which appear in the unlabeled data in at least 10 examples. We use the NLTK Snowball stemmer to stem each word of the n-grams. The smoothing hyperparameters in the computation of $P(\mathcal{D}|w)$ are set to be 1, 5, and 7 for uni-grams, bi-grams and tri-grams, respectively. We use a $\tau = 0.08$ threshold and mask an additional 5% of the training examples (in order to add noise between training epochs). We set $\tau = 0.08$ since it resulted in the successful domain alternation of more than 80% examples.

Step 2 OTT The domain classifier to get a domain-related score is based on the pretrained HuggingFace distilroberta model. It has total of 6 layers and we use the attention norm in the middle layer (layer 3, 4) to perform Over-The-Top masking

Domain Specific Prompt We choose four words representing the domain and initialize the domain-specific prompt with their word embeddings. The following is the word we use for each domain: Airline: airline, flight, seat, staff; DVD: dvd, character, actor, plot; Electronics: electronics, ipod, router, software; Kitchen: kitchen, dishwasher, pan, oven; and for the MANtIS dataset: Apple: apple, itunes, iphone, nacbook; askubuntu: askubuntu, ubuntu, apt, deb; DBA: dba, database, SQL, query; Electronics: electronics, schematics, voltage, circuit; Physics: physics, gravity, particle, quantom; Statistics: stats, regression, logits, variance;

Training of D-CONs generation model The controllable model is based on a pre-trained HuggingFace T5-base model. We train it on the unlabeled data for 20 epochs and pick the model whose generated examples for an unlabeled held-out set are of the highest domain accuracy (D.REL). 5 Training is performed with the AdamW optimizer learning rate parameter of 5e-5 and a weight decay parameter of 1e-5.

Task Classifiers Task classifiers are based on the Huggingface's RobertaForSequenceClassification model. We train the classifiers for 5 epochs with a batch size of 16 and pick the best model based on the performance on the validation set. Training is performed using the AdamW optimizer with learning rate parameters of 5e-5 for the encoder blocks and of 5e-4 for the linear layer and weight decay parameter of 1e-5.

A.2 Samples

In this section, we include masked text from each stage of our mask for our generated D-CONS.

(1) Original, **Airline**

Was traveing with my partner, we got bored halfway through the flight as there is no inflight entertainment system . hot food was served, overall a positive experience .

Step 1, **Airline** → **Electronics**

Was traveing with my partner, <m> halfway through the <m> is no <m> . <m> was <m> , <m> positive experience .

Step 2, **Airline** → **Electronics**

Was <m> with my <m>, <m> through the <m> is no <m> . <m> was <m> , <m> positive <m> .

Step 3, **Airline** → **Electronics**

Was <m> with my partner, <m> through the <m> is no <m> system. <m> was <m> , overall a positive experience .

(2) Original, **Airline**

both legs departed and arrived in time . approx 75

% full with a mix of business leisure and Disney visitors . buy on board food is reasonably priced and the pre - order Irish breakfast was delicious . fare was just €100 return . recommended .
Step 1, **Airline → kitchen**
both legs <m> and <m> in time . <m> 75 <m> a mix of <m> leisure and Disney visitors . buy <m> is reasonably priced and the <m> irish <m> was <m> . <m> was just <m> return . <m>
Step 2, **Airline → kitchen**
both <m> and <m> . <m> 75 <m> a mix of <m> and <m> . buy on <m> is reasonably priced and the <m> was <m> . <m> was just <m> return . <m>
Step 3, **Airline → kitchen**
both <m> and <m> . <m> 75 % <m> a mix of <m> and <m> . buy on <m> is resonably priced and the pre - order Irish <m> was <m> . <m> was just <m> return . recommended

(3) Original, **Dvd**
Startrek Voyager : I am a very avid " star trek " fan , and find the dvd ' s very worthwhile and interesting . (my star trek interests go back to the original series with william shatner, leonard nimoy, james doohan, et. al.
Step 1, **Dvd → Electronics**
<m> Voyager : I am a very avid <m> " <m>, and find the <m> ' s very worthwhile and <m> . (my star <m> go back to <m> with <m> , <m> nimoy , <m> doohan <m> . <m> .
Step 2, **Dvd → Electronics**
<m> : I am a very avid <m>, and <m> 's <m> worthwhile and <m> . (my <m> go back to <m> . <m> : I am a very avid " <m> " <m> , and <m> ' s <m> worthwhile and interesting . (my <m> go back to <m> .

(4) Original, **Dvd**
Tell all your friends! This is one of my favorite movies and i find it unfortunate that not too many people know about it. i think it is important for people to know this story. Everyone should watch this movie simply because it is fantastic .
Step 1, **Dvd → Kitchen** Tell all your friends! This <m> of my <m> and i find it unfortunate that not too many people know about it . i think it is important for people to know this <m> . <m> this <m> simply because it is fantastic .
Step 2, **Dvd → Kitchen** <m> friends ! This <m> of my <m> and i <m> unfortunate that not too many <m> it. i think it is important for <m> . <m> this <m> because it is <m> .
Step 3, **Dvd → Kitchen** Tell all your friends! This is one of my favorite <m> and i find it unfortunate that not too many people know about it. i think it is important for people to know this <m>. Everyone should <m> this <m> simply because it is fantastic .

(5) Original, **Electronics**
Great service, cartridge like a new one ordering was fast and easy . could not detect any inferiority in the cartridges and will order them again
Step 1, **Electronics → Airline**
Great service , cartridge like a <m> ordering was fast and easy . could not <m> any inferiority in the cartridges and will order them again .
Step 2, **Electronics → Airline**
Great <m> , <m> like <m> was fast and <m> . could not <m> any <m> in the <m> and will <m> again .
Step 3, **Electronics → Airline**
Great <m>, <m> like a new one <m> was fast and easy . could not <m> any inferiority in the <m> and will order them again

(6) Original, **Kitchen**
Conducts heat really well this set conducts heat really well and everything heats up fast but be careful not to burn anything which i did the first time using it . also it takes a bit of scrubbing if you burn it bt other than that it's a great product
Step 1, **Kitchen → Electronics**
conducts <M> really well this <M> conducts <M> really well and everything <M> up fast but be careful not to burn anything which i did the first time using it . also it takes a bit of <M> if you burn it bt other than that it ' s a great <M>
Step 2, **Kitchen → Electronics**
<M> really well this <M> really well and everything <M> up fast but <m> to <m> anything which i did the first time <m> . also it takes a bit of <M> if you <m> it <m> other than that it ' s a great <M>
Step 3, **Kitchen → Electronics**
<M> really well this <M> really well and everything <M> up fast but <m> to <m> anything which i did the first time using it . also it takes a bit of <M> if you <m> it <m> other than that it ' s a great product

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.