DoCoGen: Domain Counterfactual Generation for Low Resource Domain Adaptation

Nitay Calderon* and **Eyal Ben-David*** and **Amir Feder** and **Roi Reichart**
Technion - Israel Institute of Technology
{nitay@campus.|eyalbd12@campus.|feder@campus.|roiri@}technion.ac.il

Abstract

Natural language processing (NLP) algorithms have become very successful, but they still struggle when applied to out-of-distribution examples. In this paper we propose a controllable generation approach in order to deal with this domain adaptation (DA) challenge. Given an input text example, our DoCoGen algorithm generates a domain-counterfactual textual example (D-CON) – that is similar to the original in all aspects, including the task label, but its domain is changed to a desired one. Importantly, DoCoGen is trained using only unlabeled examples from multiple domains – no NLP task labels or parallel pairs of textual examples and their domain-counterfactuals are required. We show that DoCoGen can generate coherent counterfactuals consisting of multiple sentences. We use the D-CONs generated by DoCoGen to augment a sentiment classifier and a multi-label intent classifier in 20 and 78 DA setups, respectively, where source-domain labeled data is scarce. Our model outperforms strong baselines and improves the accuracy of a state-of-the-art unsupervised DA algorithm.[1]

1 Introduction

Natural Language Processing (NLP) algorithms are constantly improving and reaching significant milestones (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020). However, such algorithms rely on the availability of sufficient labeled data and the assumption that the training and test sets are drawn from the same underlying distribution. Unfortunately, these assumptions do not hold in many cases due to the costly and labor-intensive data labeling process and since text may originate from many different domains. As generalization in low resource regimes and beyond the training distribution are still fundamental NLP challenges, NLP algorithms significantly degrade when applied to such scenarios.

Domain adaptation (DA) is an established field of research in NLP (Roark and Bacchiani, 2003; Daumé III and Marcu, 2006; Reichart and Rappoport, 2007) that attempts to explicitly address generalization beyond the training distribution (§2). DA algorithms are trained on annotated data from source domains to be effectively applied in various target domains. Indeed, DA algorithms have been developed for multiple NLP tasks throughout the last two decades (Blitzer et al., 2006, 2007; Glorot et al., 2011; Rush et al., 2012; Ziser and Reichart, 2017, 2018a,b; Han and Eisenstein, 2019).

A natural alternative to costly human annotation would be to automatically generate labeled examples for model training. Doing so may expose the model to additional training examples and better represent the data distribution within and outside the annotated source domains. Unfortunately, generating labeled textual data is challenging (Feng et al., 2021), especially when the available labeled data is scarce. Indeed, labeled data generation has hardly been applied to DA (§2).

To allow DA through labeled data generation, we present DoCoGen, an algorithm that generates domain-counterfactual textual examples (D-CONs). In order to do that, DoCoGen intervenes on the domain-specific terms of its input example, replacing them with terms that are relevant for its target domain while keeping all other properties fixed, including the task label. Consider the task of sentiment classification (top example in Table 1). When DoCoGen encounters an example from the *Kitchen* domain (its source domain), it first recognizes the terms related to *Kitchen* reviews, i.e., *knife* and *solid*. Then, it intervenes on these terms, replacing them with text that connects the example to the *Electronics* domain (its target domain) while keeping the negative sentiment.

DoCoGen is a *controllable generation* algo-

*Both authors equally contributed to this work.
[1]Our code and data are available at https://github.com/nitaytech/DoCoGen.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7727 - 7746

rithm (Li et al., 2016; Russo et al., 2020) that is trained using a novel *unsupervised* sentence reconstruction objective. Importantly, it does not require task-annotated data, or parallel pairs of sentences and their D-CONs. A key component of DoCoGen is the *domain orientation vector*, which guides the model to generate the new text in the desired domain. The parameters of the orientation vectors are learned during the unsupervised training process, allowing the generation model to share information among the various domains it is exposed to.

We focus on two low resource scenarios: Unsupervised domain adaptation (UDA) and any domain adaptation (ADA, Ben-David et al. (2021)), with only a handful of labeled examples available from a single source domain. In both UDA and ADA the model is exposed to limited labeled source domain data and to unlabeled data from several domains. However, in UDA the *unlabeled domains* contain the future target domain to which the model will be applied, while in ADA the model has no access to the target domain during training. To cope with these extreme conditions, we use DoCoGen to enrich the source labeled data with D-CONs from the unlabeled domains. By introducing labeled D-CONs from various domains, we hope to provide the model with a training signal that is less affected by spurious correlations: Correlations between features and the task label which do not hold out-of-domain (OOD) (Veitch et al., 2021).

After a brief evaluation of the intrinsic quality of the D-CONs generated by DoCoGen, we evaluate our complete DA pipeline. We focus on two tasks: Binary sentiment classification of reviews and multi-label intent prediction in information-seeking conversations. In both tasks, we follow the UDA and ADA scenarios, for a total of 12 and 8 sentiment setups, respectively, as well as 30 UDA and 48 ADA intent prediction setups. Our results demonstrate the superiority of DoCoGen over strong DA and textual-data augmentation algorithms. Finally, combining DoCoGen with PERL (Ben-David et al., 2020), a SOTA UDA model, yields new SOTA DA accuracy and stability.

2 Related Work

We first describe research in our DA setups: UDA and ADA. We then continue with the study of counterfactual-based data augmentation, and, finally, we describe research on counterfactual generation methods.

Domain Adaptation (DA) The NLP literature contains several DA setups, the most realistic of which is *unsupervised domain adaptation* (UDA), which assumes the availability of unlabeled data from a source and a target domain, as well as access to labeled data from the source domain (Blitzer et al., 2006). An even more challenging and potentially more realistic setup is the recently proposed *any domain adaptation* setup (ADA, Ben-David et al. (2021)), which assumes no knowledge of the target domains at training time. There are several approaches to DA, including representation learning (Blitzer et al., 2006; Ziser and Reichart, 2017) and data-centric approaches like instance re-weighting and self-training (Huang et al., 2006; Rotman and Reichart, 2019).

Since the rise of deep neural networks (DNNs), most focus in DA research has been directed to deep representation learning approaches (DReL). One line of DReL work employs an input reconstruction objective (Glorot et al., 2011; Chen et al., 2012; Yang and Eisenstein, 2014; Ganin et al., 2016). Another line employs pivot features, which are prominent to the task of interest and common in the source and target domains (Blitzer et al., 2007; Pan et al., 2010; Ziser and Reichart, 2018b; Ben-David et al., 2020; Lekhtman et al., 2021).

We deviate from the DReL approach to DA and propose a data-centric methodology. Contrary to the above works, our approach can be applied to both UDA and ADA. Moreover, unlike previous ADA work, which builds upon multi-source DA, our approach can also perform single-source ADA.

Counterfactually Augmented Data (CAD) Textual data augmentation (TDA) is a technique for increasing the training dataset without explicitly collecting new examples. This is achieved by adding slightly modified copies of already existing examples (local sampling) or newly created data (global sampling). TDA serves as a solution for insufficient data scenarios and as a technique for improving model robustness (Xie et al., 2020; Ng et al., 2020). There are rule-based and model-based approaches to TDA. Rule-based methods commonly involve insertion, deletion, swap and replacement of specific words (Wei and Zou, 2019), or template-based paraphrasing (Rosenberg et al., 2021). Model-based methods typically utilize a pretrained language model (PLM), e.g., for replacing random words (Kobayashi, 2018; Ng et al., 2020), or generating entirely new examples

Original, **Kitchen**: A good knife but Quality Control was poor. The knife is solid and very comfortable in hand, however, when I got it new, the blade is slightly bent. I expect it to be in almost perfect condition, but it's not.
DoCoGen, **Kitchen** → **Electronics**: A good product but Quality Control was poor. The ipod is very easy to use and very comfortable in hand, however, when I got it new, the ipod is slightly flimsy. I expect it to be in almost perfect shape, but it's not.
Original, **DVD**: The direction of this film is excellent. I love all the characters and the way they interact. The storyline is very important also. It's about religious beliefs and neighbors that interact with each other. It's a well-paced and interesting story that's not like anything else I've ever seen.
DoCoGen, **DVD** → **Airline**: The service on this flight is excellent. I love the staff and the way they interact. The safety is very important also. It's nice to have staff and neighbors that can help each other. It's a well-groomed and professional crew that's not like anything else I've ever experienced.
Original, **Electronics**: That relay board is only good for switching AC loads of 100V or more. If you have a lower voltage load, it's not going to work. For low voltage loads use transistors, MOSFETs or a ULN2803 driver board.
DoCoGen, **Electronics** → **Statistics**: That model is only good for data of n or more. If you have a lower n, it's not going to work. For lower n regression use a linear, logistic or a t-test.

Table 1: Domain-counterfactual textual examples (D-CONs) generated by DoCoGen. Red terms are replaced with green terms through the process of D-CON generation. For additional examples see §A.

from a prior data-distribution (Bowman et al., 2016; Russo et al., 2020; Wang et al., 2021). Other model-based methods apply backtranslation (Edunov et al., 2018) or paraphrasing (Kumar et al., 2019) for local sampling.

Another approach within local sampling TDA is to change (only) a specific concept that exists in the original example, creating a counterfactual example. Counterfactually-Augmented Data (CAD) is generated by minimally intervening on examples to change their ground-truth label, that is, perturbing only those terms necessary to change the label (Kaushik et al., 2020). CAD is commonly used to improve generalizability (Kaushik et al., 2020; Sen et al., 2021), however empirical results using CAD for OOD generalization have been mixed (Joshi and He, 2021; Khashabi et al., 2020).

In this work, we explore a different type of counterfactuals, namely D-CONs, which are the result of intervening only on the example's domain while holding everything else equal, particularly its task label. For sentiment analysis, we may be, for example, interested in revising a negative movie review, making it a negative airline review. In addition, while CAD is mostly generated via a human-in-the-loop process (Kaushik et al., 2020; Khashabi et al., 2020; Sen et al., 2021), our work focuses on automatic counterfactual generation.

Counterfactual Generation *controllable generation* refers to generation of text while controlling for specific attributes (Prabhumoye et al., 2020). The controlled attributes can range from style (e.g., politeness and sentiment) to content (e.g., keywords and entities) and even topic. Keskar et al. (2019) propose to control the generated text by training an LM on datasets annotated with the controlled attributes, and Meister et al. (2020) modify the model's decoding method. Recently, Russo et al. (2020) introduced a global sampling conditional variational autoencoder (VAE), augmenting text while controlling for attributes such as label and verb tense. However, controlling for the task label is challenging in scarce labeled data scenarios (Chen et al., 2021), since generative models require large amounts of labeled data .

Counterfactual generation lies at the intersection of controllable generation and causal inference (Feder et al., 2021a). Only few works deal with counterfactual generation, mostly by intervening on the task label. Wu et al. (2021) train a model on textual examples and their manually generated counterfactuals. Other works present methods for controlling for the text domain and semantics (Wang et al., 2020; Feng et al., 2019), yet they all experiment with short texts, while our model can generate longer texts, consisting of multiple sentences. A recent work by Yu et al. (2021) focuses on generation of new target-domain examples for aspect-based sentiment analysis (ABSA) (Pontiki et al., 2016). However, this method is designed specifically for ABSA, utilizing predefined knowledge, and is only suitable for UDA setups where source domain labeled data is abundant. Our work presents a novel domain counterfactual generation algorithm, which can be trained in an unsupervised manner, and its generated outputs are demonstrated to be effective in multiple low-resource DA tasks.

3 Domain-Counterfactual Examples

In this section, we formally define the concept of domain-counterfactual textual examples (D-CONs) and discuss the motivation behind them.

Definition x' is a *domain-counterfactual example* (D-CON) of x if it is a coherent human-like text that is a result of intervening on the domain of x and changing it to another domain, while holding everything else equal. Particularly, we would like the task label of x' and x to be identical. Formally, given an example $(x, y) \sim \mathcal{D}$ and a destination domain $\mathcal{D}'$, the goal of D-CON generation is to generate $x' \sim P_{\mathcal{D}'}(X|Y = y)$ such that $x' \simeq_{\mathcal{D}'} x$, where $\simeq_{\mathcal{D}'}$ is the domain counterfactual operator.

In this work, given a labeled source example x we aim to generate coherent human-like D-CONs from the unlabeled domains (see §1). We propose a D-CON generation algorithm, DoCoGen, consisting of two components. The first involves masking domain specific terms of the given example, yielding $\texttt{M}(x)$. The second is a controllable generation model G which takes as input $\texttt{M}(x)$ and a *domain orientation vector* v'. This vector specifies the destination domain $\mathcal{D}'$, controlling the semantics of the generated D-CON. Formally:

$$\texttt{DoCoGen}(x, \mathcal{D}') = \texttt{G}(\texttt{M}(x), v') \simeq_{\mathcal{D}'} x$$

Motivation The NLP community has recently become increasingly concerned with *spurious correlations* (Geirhos et al., 2020; Wang and Culotta, 2020; Gardner et al., 2021). In the case of DA, spurious correlations may be defined as correlations between X and Y which are relevant only to a specific domain or in a certain sample of labeled examples. Such correlations may make a predictor $f : \mathcal{X} \rightarrow \mathcal{Y}$ brittle to domain shifts.

Using counterfactuals w.r.t. a specific variable allows us to both estimate its effect on our predictor (Feder et al., 2021b; Rosenberg et al., 2021) or alleviate its impact on it (Kaushik et al., 2021). We focus on the latter, automatically generating D-CONs by intervening on the domain variable $\mathcal{D}$. Adding these D-CONs to the training set of a predictor should reduce its reliance on domain-specific information and spurious correlations.

From a DA perspective, enriching the training data with D-CONs is motivated by pivot features (§2), which are frequent in multiple domains and are prominent for the task. D-CONs preserve language patterns, such as pivots, which are frequent in multiple domains. Consider the middle example in Table 1, pivot words (such as *excellent* and *important*) are preserved in the D-CON, while non-pivots (*intereseting* and *well-paced*) are replaced due to the domain intervention. Accordingly, a model trained on an example and its D-CON is directed to focus on pivots rather than on non-pivots, consequently generalizing better OOD.

4 DoCoGen: Domain Counterfactual Generation

We propose a corrupt-and-reconstruct approach for generating D-CONs from given source domain examples (Figure 1). We next extend on these two steps, and describe our filtering mechanism used to disqualify low quality D-CONs.

4.1 Domain Corruption

The first step of generating a D-CON is to mask domain specific terms. In order to mask an example $x \sim \mathcal{D}$ with a destination domain $\mathcal{D}'$, we first mask all uni-grams w with $\texttt{m}(w, \mathcal{D}, \mathcal{D}') > \tau$, where τ is a hyperparameter and m is a masking score that is defined later in this section. Then, we mask all the remaining bi-grams (that do not contain a masked uni-gram) according to the same masking threshold τ. This process is repeated up to tri-gram expressions. The final output of the corruption step is a masked example $\texttt{M}(x)$.

In Figure 1, the masking scores of uni-grams and bi-grams appear above the input words. An n-gram is masked if and only if its score is above a $\tau = 0.08$ threshold and the scores of its grams are lower. For example, *system* is not masked although the bi-gram *entertainment system* has a score above the τ threshold, since *entertainment* is masked and the score of *system* is lower than τ.

Masking Score Let w be an n-gram and $\mathcal{D}$ be a domain with $n_{\mathcal{D}}$ unlabeled examples. We denote the number of examples from $\mathcal{D}$ that contain w by $\#_{w|\mathcal{D}}$. By assuming that domains have equal prior probabilities and by using the Bayes' rule, the probability of $\mathcal{D}$ given w can be estimated by $P(D = \mathcal{D}|W = w) \propto \frac{\#_{w|\mathcal{D}}+\alpha}{n_{\mathcal{D}}}$, where α is a smoothing hyperparameter. We define the affinity of w to $\mathcal{D}$ to be:

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

where N is the number of unlabeled domains and $H(D|w)$ is the entropy of $D|w$, which is upper

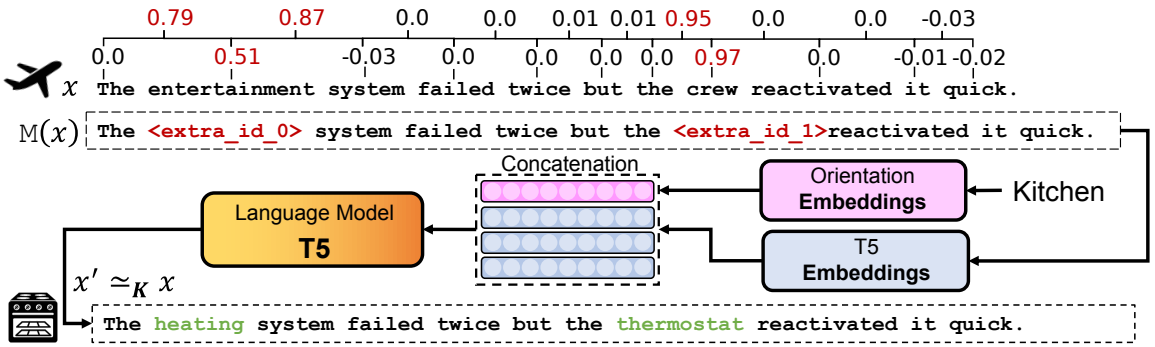

Figure 1: The DoCoGen model. Given a review x from the *airline* domain, we aim to generate a D-CON from the *kitchen* domain. We first corrupt the domain of the example by masking domain specific terms. The numbers above the input words are the masking scores of uni-grams and bi-grams. Terms with scores above a threshold ($\tau = 0.08$) are masked. In the reconstruction step we use a T5-based generation model to generate the D-CON $x' \simeq_{\mathbf{K}} x$. The input of the model is a concatenation of the *orientation vector* that represents the target domain with the model's embedding vectors which correspond to the tokens of the masked example $\mathtt{M}(x)$.

bounded by $\log N$. Notice that higher $H(D|w)$ values indicate that w is not related to any specific domain. Finally, we set the masking score of an n-gram w with an origin domain $\mathcal{D}$ and a destination domain $\mathcal{D}'$ as follows:

$$\mathtt{m}(w, \mathcal{D}, \mathcal{D}') = \rho(w, \mathcal{D}) - \rho(w, \mathcal{D}')$$

Note that $\mathtt{m}(w, \mathcal{D}, \mathcal{D}') \in [-1, 1]$. It can be negative due to the right hand side's subtrahend, which aims to prevent masking n-grams that are related to the destination domain and should appear in the counterfactual, like *system* in Figure 1.

4.2 Domain-Oriented Reconstruction

The second step of DoCoGen is a reconstruction step that involves a generative model, based on an encoder-decoder T5 architecture (Raffel et al., 2020). Given a masked example $\mathtt{M}(x)$ and a destination domain $\mathcal{D}'$, we concatenate a domain orientation vector v' that represents $\mathcal{D}'$ with the masked input's embedding vectors. Then, the concatenated matrix is passed as an input to the encoder-decoder model for counterfactual generation, yielding x'. We next describe the mechanism behind domain orientation vectors.

Domain Orientation Vectors In addition to the T5 embedding matrix (T5 Embeddings in Figure 1), we equip our model with another learnable embedding matrix, containing $K \cdot N$ orientation vectors, such that each domain is represented by K different vectors (Orientation Embeddings in Figure 1). We initialize the orientation vectors with the T5 embedding vectors of the domain names and the top $K-1$ *representing words* of each domain. The top representing words of domain $\mathcal{D}$ are those which reach the highest score of: $\log(\#_{w|\mathcal{D}} + 1)\rho(w, \mathcal{D})$. We use K orientation vectors to allow us generate a heterogeneous set of D-CONs for a given destination domain (see examples in §A). We note that although the orientation vectors are initialized with vectors from the T5 embedding matrix, they have a different role and thus are likely to converge to different values during the training process.

Training In the spirit of low resource learning, we would like to train DoCoGen in an unsupervised manner, i.e., without access to manually generated D-CONs. Therefore, we use the unlabeled data of our unlabeled domains. For each example x, we provide the model with $\mathtt{M}(x)$, the corrupted version of x, and v, the orientation vector of $\mathcal{D}$, and with x as the gold output. The model hence learns to reconstruct x given $\mathtt{M}(x)$ and v.

Notice that the origin and the destination domains are the same, i.e, $\mathcal{D} = \mathcal{D}'$, and the masking score is $\mathtt{m}(w, \mathcal{D}, \mathcal{D}) = 0$. Hence, for masking purposes, we randomly choose $\tilde{\mathcal{D}} \neq \mathcal{D}$ and plug it as the destination domain in the masking score. We then choose an orientation v for $\mathcal{D}$, by randomly sampling either the domain name or one of its representing words as long as it appears in x.

Finally, since the orientation vector parameters are trained as part of the reconstruction objective, we establish the connection between the orientation vector and the semantics of the completed example. Hence, we expect that at inference time examples will be properly transformed into their D-CONs.

Inference Given $(x, \mathcal{D}, \mathcal{D}')$, we first mask the example to get $\texttt{M}(x)$ and select one orientation vector v' that represents $\mathcal{D}'$.[2] Together, the tuple $(\texttt{M}(x), v')$ forms the input, and accordingly the model generates a D-CON $x' \simeq_{\mathcal{D}'} x$. To increase the likelihood that x' originates from $\mathcal{D}'$, we restrict the model to generate only tokens of the original example or tokens that are related to $\mathcal{D}'$ and meet the condition: $\max_{i \in 1,\ldots,N} \texttt{m}(w, \mathcal{D}', \mathcal{D}_i) > \tau$.

4.3 Filtering Mechanism

In order to properly apply DoCoGen within a DA pipeline, we introduce a filtering mechanism that disqualifies low quality D-CONs generated by DoCoGen. Particularly, we train a classifier to predict the domain of the original, human-written unlabeled examples, and use it to remove D-CONs if their predicted domain is not the given destination domain. In addition, we disqualify D-CONs with less than four words or when the word overlap with the original example is lower than 25%. We name DoCoGen when equipped with this filtering mechanism F-DoCoGen.

5 Intrinsic Evaluation

We next assess DoCoGen in terms of its generated D-CONs, ensuring they: (i) belong to the correct domain and label (1, 2), and (ii) are fluent (3, 4). To this end, we collected 20 original reviews, equally distributed among four domains (the A, D, E, and K domains, see §6). We then applied DoCoGen to generate 60 D-CONs, 3 for each of the original reviews (see §6 for the DoCoGen training setup). Finally, we trained the VAE model of Russo et al. (2020) on labeled data (all the labeled data of the A, D, E, and K domains) and applied it to generate five reviews from each of the above four domains, with the same number of positive and negative reviews as in the set of original reviews.

We then conducted a crowd-sourcing experiment where five nearly native English speakers rated each example, considering the following evaluation measures: (1) Domain relevance (D.REL) - whether the topic of the generated text is related to its destination domain; (2) Label preservation (L.PRES) - what is the label of the generated example (and we report whether the answer was identical to the desired label); (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 that have to be performed to make the example logical and grammatical.[3]

	↑D.REL	↑L.PRES	↑ACCPT	↓WER
VAE	90.0	46.0	2.11	0.54
DoCoGen	93.0	80.0	4.01	0.17
Original Reviews	99.0	88.0	4.73	0.10

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

Table 2 reports our results. DoCoGen achieves high ACCPT scores and low WER scores, significantly outperforming its VAE alternative, which is known to struggle with longer texts (Shen et al., 2019; Iqbal and Qureshi, 2020). Interestingly, DoCoGen achieves compatible results to the original reviews, indicating the high quality of its generated texts. Finally, in more than 90% of the cases DoCoGen manages to change the example domain to the desired domain, and in 80% it preserves the original example label. In comparison, only 88% of the original examples were annotated as their gold label.

6 Experimental Setup

6.1 Tasks and Domains[4]

In this subsection we describe our tasks and datasets, as well as the two DA setups which are the focus of this work. A full description of the number of samples in each dataset is found in Table 6.

Sentiment Classification We follow a large body of prior DA work, focusing on the task of binary sentiment classification. Specifically, our experiments include six different domains: the four legacy product review domains (Blitzer et al., 2007) - Books (B), DVDs (D), Electronic items (E) and Kitchen appliances (K); the challenging airline review dataset (A) (Nguyen, 2015; Ziser and Reichart, 2018b); and the restaurant (R) domain obtained from the Yelp dataset challenge (Zhang et al., 2015). The focus of this work is on low resource DA, and thus we randomly sample 100 labeled examples to form the training set for the following domains: A, D, E, and K.

As described in §2, we explore two DA setups, UDA and ADA. For UDA, where the model has

[2] §B.3 presents the % of masked tokens in our experiments.

[3] We actually asked the annotators to edit the example and then measured the number of edit operations.

[4] URLs of the datasets and the code, implementation and hyperparameter details are described in §B.

access to unlabeled target domain data, we experiment with 12 cross-domain setups, including the following domains: A, D, E, and K. For ADA, where unlabeled data from the target domain is not within reach, we experiment with a total of 8 setups, including B and R as target domains, and A, D, E, and K as source domains. Our reported accuracy scores are averaged across 25 different seeds and randomly sampled training and development sets.

Multi-label intent prediction Our second task is multi-label intent prediction of utterances from information-seeking conversations. We use the multi-domain MANtIS dataset (Penha et al., 2019), consisting of diverse conversations from the question-answering Stack Exchange portal. The authors provide manually annotated user intent utterances, with eight possible intent labels, such as *information request*, *potential answer* and *greetings*. Since we focus on low resource scenarios, we use only the five most common labels, as the frequency of the other three labels is less than 5%, and in some domains they are completely missing.

The MANtIS dataset consists of 14 domains: Apple (AP), DBA (DB), Electronics (EL), Physics (PH), Statistics (ST), askubuntu (UB); DIY (DI), English (EN), Gaming (GA), GIS (GI), Sci-Fi (SC), Security (SE), Travel (TR) and Worldbuilding (WO). We use the first 6 domains as unlabeled domains, randomly sampling train, development and test sets for each. The remaining 8 domains are used as target domains in the ADA setup, resulting in 30 UDA (6×5) and 48 ADA (6×8) setups.

Following Penha et al. (2019), we use the (Macro) F1-score to measure classifier performances, and, like in the sentiment classification task, our reported results are averaged across 25 different seeds and randomly sampled training sets.

DA by Augmentation The DA pipeline includes a T5-based sentiment classifier trained on labeled data from a single source domain and an augmentation model (e.g., DoCoGen) trained on unlabeled data from four unlabeled domains. We first train DoCoGen on the unlabeled data, and then use it for generating D-CONs that enrich the classifier's training data. For each labeled training example, DoCoGen generates $K = 4$ D-CONs w.r.t. each unlabeled domain, resulting in a total of 16 D-CONs per example. After training the sentiment classifier on the enriched data, we evaluate it on test examples originating from one of the unlabeled domains (UDA) or one of the unseen domains (ADA). We denote each DA model by the algorithm that was used for enriching its training data.

6.2 Models and Baselines

Our main models are DoCoGen and F-DoCoGen, which is equipped with the filtering mechanism. We compare them to three types of models: (a) baseline models, including both baselines for the entire DA pipeline (1,2,5) and alternative augmentation methods (3,4); (b) ablation models (6,7) that use variants of our D-CON generation algorithm where one component is modified, highlighting the importance of our design choices; and (c) an upper-bound generation model that has access to labeled data from the target domains. Unless otherwise stated, all sentiment classifiers use the same architecture, based on a pre-trained T5 model. We next describe the models in each of these groups.

Baseline DA Models We experiment with five baselines: (1) *No-Domain-Adaptation* (NoDA), A model that is only trained on the available training data from the source domain in each DA setup; (2) *Domain-Adversarial-Neural-Network* (DANN), A model that integrates the sentiment analysis predictive task with an adversarial domain classifier to learn domain invariant representations (Ganin et al., 2016). This model does not apply augmentation, but instead the unlabeled data is used for training its adversarial component; (3) *Easy-Data-Augmentation (EDA)*, an augmentation method that randomly inserts, swaps, and deletes words or replaces synonyms (Wei and Zou, 2019); (4) *Random-masking Random-Reconstructing* (RM-RR), another basic augmentation method that randomly masks tokens from the input example and then fills the masks with tokens that are chosen by a masked language modeling head, as suggested by (Ng et al., 2020); and (5) *PERL*, a SOTA model for the UDA setup (Ben-David et al., 2020).

Ablation Models We consider two variants of DoCoGen: (6) *No-Orientation-Vectors* (No-OV), a generation model that masks tokens by employing a similar masking mechanism as DoCoGen, and then employing a masked language modeling head to fill the masked tokens (without domain orientation vectors); and (7) *Random-Masking with Orientation-Vectors* (RM-OV), a generation model that randomly masks tokens from the input example and then employs the DoCoGen's reconstruction mechanism to fill the masks.

	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
DANN	70.3	78.7	78.9	75.5	81.2	82.3	82.3	78.3	86.7	81.0	78.3	85.0	79.9
EDA	69.3	79.1	79.4	71.1	79.9	83.0	79.9	80.8	88.0	75.7	80.9	86.4	79.5
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
No-OV	67.2	76.5	76.1	71.5	79.7	82.9	80.9	80.5	88.9	74.8	79.6	85.3	78.7
RM-OV	69.3	**80.2**	**80.4**	72.7	81.8	84.5	79.6	81.7	89.0	70.3	79.4	85.4	79.5
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
F-DoCoGen	**71.1**	79.6	79.6	**76.7**	**83.2**	**84.8**	82.6	**82.1**	89.2	**81.4**	**83.3**	**88.0**	**81.8**
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-PERL	75.7	82.7	83.1	82.4	85.0	84.9	81.3	80.8	88.3	79.5	80.9	86.2	82.6
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: Sentiment classification: accuracy scores for each source and target domain pair in the UDA setup. **Bold** numbers mark the best performing T5-based model, and underline numbers mark the best performing PERL model.

Source	A		D		E		K		
Target	B	R	B	R	B	R	B	R	AVG
NoDA	69.1	76.5	82.3	82.8	81.5	84.5	82.4	85.2	80.5
DANN	70.5	77.2	82.7	81.5	80.9	83.4	81.8	83.4	80.2
EDA	69.3	78.0	83.7	82.6	83.2	85.4	82.8	86.3	81.4
RM-RR	69.4	78.4	83.8	83.5	81.9	85.6	83.7	85.4	81.5
No-OV	67.1	76.1	83.8	82.5	82.9	**86.2**	83.0	85.6	80.9
RM-OV	69.6	78.7	84.3	**83.6**	83.6	**86.2**	83.9	85.5	81.9
DoCoGen	70.9	78.1	84.4	82.9	83.9	86.0	84.5	85.7	82.1
F-DoCoGen	**71.4**	**79.3**	**84.9**	**83.6**	**84.2**	86.1	**85.6**	**87.2**	**82.8**
Oracle-Gen	84.4	85.2	86.7	86.1	86.0	86.5	85.3	86.5	85.8

Table 4: Sentiment classification: accuracy scores for each source and target domain pair in the ADA setup.

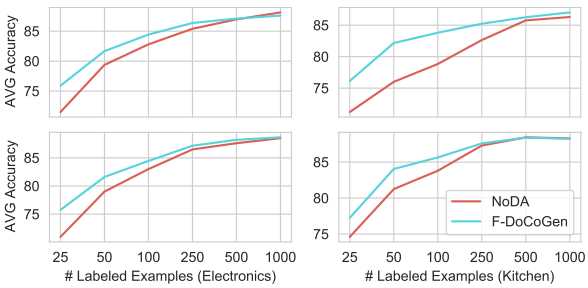

Figure 2: Average accuracy in UDA (top) and ADA (bottom) setups with different number of labeled examples from two source domains: E and K.

Upper-Bound We implement an upper-bound model for D-CON augmentation, *Oracle-Matching* (Oracle-Gen). Unlike all other models in this work, Oracle-Gen has access to target domain labeled data. Thus, given an example from a source domain, Oracle-Gen looks for the most similar example with the same label in the target domain, and adds it to its training data (see §B.1).

7 Results

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 all target domains, in both UDA and ADA.

D-CON Generation Impact For sentiment classification, our model, F-DoCoGen, outperforms all baseline models (NoDA, DANN, EDA, and RM-RR) in 10 of 12 UDA setups and in all ADA setups, exhibiting average performance gains of 1.9% and 1.3% over the best performing baseline model in the UDA (DANN) and the ADA (RM-RR) setups, respectively. Moreover, DoCoGen without filtering, is also superior to all baselines, reaching average gains of 1.6% and of 0.6% across all UDA and ADA setups, respectively. For intent prediction, DoCoGen (without filtering) is the best performing model, outperforming all baselines across all setups, and reaching average gains of 1.6% and 1.5% across all UDA and ADA setups, respectively. Since many intent examples are not domain-specific, our filtering mechanism tends to easily remove their DoCoGen generated D-CONs. We believe that this is the reason for the small degradation in F-DoCoGen performance compared to DoCoGen. However, F-DoCoGen still consistently outperforms all baselines. These results highlight the impact of D-CON generation on model robustness in low-resource setups. Finally, our models are also stable: Their std is lower than all baselines (see §C.1).

Ablation Models The tables further demonstrate that F-DoCoGen outperforms its ablation models (§ 6.2), namely No-OV and RM-OV, in 10 of 12 and 7 of 8 UDA and ADA sentiment classification setups, respectively, and the same holds for DoCoGen across all intent prediction setups. Furthermore, in sentiment classification, F-DoCoGen achieves an average error reduction of 11.2% and 5.0% in UDA and ADA, respectively, over the strongest ablation model (RM-OV), while in intent prediction DoCoGen achieves a reduction of 8% and 7.6%, in both setups, respectively. Fi-

Source	AP		DB		EL		PH		ST		UB		AVG	
Setup	**UDA**	**ADA**	**UDA**	**ADA**	**UDA**	**ADA**	**UDA**	**ADA**	**UDA**	**ADA**	**UDA**	**ADA**	**UDA**	**ADA**
NoDA	75.5	74.3	72.2	71.0	71.2	70.8	67.1	67.0	71.8	70.0	72.0	71.1	71.6	70.7
DANN	76.1	75.3	73.7	73.1	72.8	72.5	72.6	72.0	74.6	72.8	72.8	72.8	73.8	73.1
EDA	71.5	70.3	69.5	67.7	69.3	68.7	65.1	64.6	70.1	68.9	69.7	68.0	69.2	68.0
RM-RR	75.3	74.3	72.8	71.3	72.3	71.7	67.4	67.5	72.9	71.2	73.0	71.8	72.3	71.3
No-OV	76.5	75.3	73.5	72.4	72.7	72.6	69.9	70.3	73.6	72.2	73.3	72.3	73.2	72.5
RM-OV	75.0	74.4	72.5	71.0	72.2	72.3	69.9	70.1	72.3	71.3	73.2	72.3	72.5	71.9
DoCoGen	**77.5**	**76.5**	**75.0**	**74.0**	**74.5**	**74.2**	**74.6**	74.1	**76.3**	74.6	**74.8**	74.1	**75.4**	**74.6**
F-DoCoGen	76.9	76.2	74.6	73.3	73.7	73.2	**74.6**	**74.6**	**76.3**	**74.8**	74.5	**74.2**	75.1	74.4
Oracle-Gen	80.7	80.5	79.6	79.3	78.4	78.8	79.8	79.7	80.4	79.2	81.0	80.5	80.0	79.7

Table 5: Intent prediction: F1 scores for UDA and ADA intent prediction. We report the average F1 score across five or seven target domains (UDA and ADA setups respectively).

nally, our results demonstrate the importance of inappropriate D-CONs disqualification, as in the task of sentiment classification, F-DoCoGen outperforms DoCoGen in 8 of 12 UDA setups and in all ADA setups. On the other hand, when non domain-specific examples are frequent, filtering might lead to small performance degradation, as happens in the intent prediction task. Our results hence stress the importance of each of DoCoGen's algorithmic components, i.e. *domain-corruption* (§ 4.1 F-DoCoGen vs RM-OV) and *oriented-reconstruction* (§ 4.2 F-DoCoGen vs No-OV).

Complementary Effect with SOTA Models We notice that F-DoCoGen replicates the average performance of PERL (Ben-David et al., 2020), the UDA SOTA, in sentiment classification. However, since PERL is based on a different architecture than the rest of the models (BERT vs T5), the models are not directly comparable. PERL is a pivot-based representation learning method for DA, which applies pre-training on unlabeled target data and is hence relevant only for UDA. Since DoCoGen implements a different approach to DA (D-CON generation), we check for the complementary effect of these models: DoCoGen-PERL first augments the labeled data with D-CONs and then continues with the PERL pipeline. As reported in Table 3, DoCoGen-PERL outperforms PERL in 8 of 12 UDA setups, providing an average improvement of 0.7%. Furthermore, the average std of DoCoGen-PERL is 2.1 compared to 3.6 of PERL (§C.1). This stresses the stability of DoCoGen-PERL across these challenging setup (Ziser and Reichart, 2019).

Unfortunately, we cannot perform an equivalent comparison in the ADA setup, since its SOTA models (Ben-David et al., 2021; Wright and Augenstein, 2020) employ labeled data from multiple sources. Likewise, since PERL is not designed for multi-label prediction, we could not apply it to intent prediction. To the best of our knowledge, we are the first to effectively perform single-source ADA.

Training Size Effect We would next like to understand the effect of D-CONs generated by DoCoGen on classifiers trained with manually labeled training sets of various sizes. Figure 2 shows that the effect of D-CON augmentation vanishes when the unaugmented classifier reaches accuracy above 85% and a performance plateau (visualized as an elbow in the curve). These results support our hypotheses that low-resource DA scenarios may result in a model that latch on spurious domain correlations, impeding its performance. Accordingly, generating D-CONs by intervening on the domain essentially reduces the reliance on domain-specific information and spurious correlations.

8 Conclusions

We presented DoCoGen, a corrupt-and-reconstruct approach for generating domain-counterfactuals (D-CONs) and apply it as a data augmentation method in low-resource DA. We hypothesized that D-CONs may mitigate the reliance on domain-specific features and on spurious correlations and help generalize out of domain.

Our augmentation strategy yields robust models that outperform strong baselines across many low-resource DA setups. In future work we would like to further improve the controllable generation quality of DoCoGen, potentially extending it to control for multiple attributes. Moreover, we would like our methodology to address additional NLP tasks and DA setups.

Acknowledgements

We would like to thank the action editor and the reviewers, as well as the members of the IE@Technion NLP group for their valuable feedback and advice. This research was partially funded by an ISF personal grant No. 1625/18.

References

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2021. PADA: A prompt-based autoregressive approach for adaptation to unseen domains. *CoRR*, abs/2102.12206.

Eyal Ben-David, Carmel Rabinovitz, and Roi Reichart. 2020. PERL: pivot-based domain adaptation for pre-trained deep contextualized embedding models. *Trans. Assoc. Comput. Linguistics*, 8:504–521.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

John Blitzer, Ryan T. McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP 2006, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 22-23 July 2006, Sydney, Australia*, pages 120–128. ACL.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 10–21. ACL.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2021. An empirical survey of data augmentation for limited data learning in NLP. *CoRR*, abs/2106.07499.

Minmin Chen, Zhixiang Eddie Xu, Kilian Q. Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500. Association for Computational Linguistics.

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. 2021a. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *CoRR*, abs/2109.00725.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021b. Causalm: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 968–988. Association for Computational Linguistics.

Steven Y. Feng, Aaron W. Li, and Jesse Hoey. 2019. Keep calm and switch on! preserving sentiment and fluency in semantic text exchange. 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 2701–2711. Association for Computational Linguistics.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky.

2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17:59:1–59:35.

Matt Gardner, William Merrill, Jesse Dodge, Matthew E. Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. Competency problems: On finding and removing artifacts in language data. *CoRR*, abs/2104.08646.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *CoRR*, abs/2004.07780.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 513–520. Omnipress.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. 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 4237–4247. Association for Computational Linguistics.

Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. 2006. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 601–608. MIT Press.

Touseef Iqbal and Shaima Qureshi. 2020. The survey: Text generation models in deep learning. *Journal of King Saud University-Computer and Information Sciences*.

Nitish Joshi and He He. 2021. An investigation of the (in)effectiveness of counterfactually augmented data. *CoRR*, abs/2107.00753.

Divyansh Kaushik, Eduard H. Hovy, and Zachary Chase Lipton. 2020. Learning the difference that makes A difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Divyansh Kaushik, Amrith Setlur, Eduard H. Hovy, and Zachary Chase Lipton. 2021. Explaining the efficacy of counterfactually augmented data. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. 2020. More bang for your buck: Natural perturbation for robust question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 163–170. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 452–457. Association for Computational Linguistics.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha P. Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3609–3619. Association for Computational Linguistics.

Entony Lekhtman, Yftah Ziser, and Roi Reichart. 2021. DILBERT: customized pre-training for domain adaptation withcategory shift, with an application to aspect extraction. *CoRR*, abs/2109.00571.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2173–2185. Association for Computational Linguistics.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. SSMBA: self-supervised manifold based data augmentation for improving out-of-domain robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*,

pages 1268–1283. Association for Computational Linguistics.

Quang Nguyen. 2015. The airline review dataset.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 751–760. ACM.

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

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia V. Loukachevitch, Evgeniy V. Kotelnikov, Núria Bel, Salud María Jiménez Zafra, and Gülsen Eryigit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 19–30. The Association for Computer Linguistics.

Shrimai Prabhumoye, Alan W. Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 1–14. International Committee on Computational Linguistics.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3980–3990. Association for Computational Linguistics.

Brian Roark and Michiel Bacchiani. 2003. Supervised and unsupervised PCFG adaptation to novel domains. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics.

Daniel Rosenberg, Itai Gat, Amir Feder, and Roi Reichart. 2021. Are VQA systems rad? measuring robustness to augmented data with focused interventions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 61–70. Association for Computational Linguistics.

Guy Rotman and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Trans. Assoc. Comput. Linguistics*, 7:695–713.

Alexander M Rush, Roi Reichart, Michael Collins, and Amir Globerson. 2012. Improved parsing and pos tagging using inter-sentence consistency constraints. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1434–1444.

Giuseppe Russo, Nora Hollenstein, Claudiu Cristian Musat, and Ce Zhang. 2020. Control, generate, augment: A scalable framework for multi-attribute text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 351–366. Association for Computational Linguistics.

Indira Sen, Mattia Samory, Fabian Flöck, Claudia Wagner, and Isabelle Augenstein. 2021. How does counterfactually augmented data impact models for social computing constructs? *CoRR*, abs/2109.07022.

Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Jianfeng Gao, and Lawrence Carin. 2019. Towards generating long and coherent text with multi-level latent variable models. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2079–2089. Association for Computational Linguistics.

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

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020. Cat-gen: Improving robustness in NLP models via controlled adversarial text generation. In *Proceedings of the 2020 Conference on Empirical Methods*

in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 5141–5146. Association for Computational Linguistics.

Zhao Wang and Aron Culotta. 2020. Identifying spurious correlations for robust text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 3431–3440. Association for Computational Linguistics.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021. Towards zero-label language learning. *CoRR*, abs/2109.09193.

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification tasks. 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 6381–6387. Association for Computational Linguistics.

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, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Dustin Wright and Isabelle Augenstein. 2020. Transformer based multi-source domain adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7963–7974. Association for Computational Linguistics.

Tongshuang Wu, Marco Túlio Ribeiro, Jeffrey Heer, and Daniel S. Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6707–6723. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yi Yang and Jacob Eisenstein. 2014. Fast easy unsupervised domain adaptation with marginalized structured dropout. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 538–544. The Association for Computer Linguistics.

Jianfei Yu, Chenggong Gong, and Rui Xia. 2021. Cross-domain review generation for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4767–4777. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yftah Ziser and Roi Reichart. 2017. Neural structural correspondence learning for domain adaptation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 400–410. Association for Computational Linguistics.

Yftah Ziser and Roi Reichart. 2018a. Deep pivot-based modeling for cross-language cross-domain transfer with minimal guidance. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 238–249. Association for Computational Linguistics.

Yftah Ziser and Roi Reichart. 2018b. Pivot based language modeling for improved neural domain adaptation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1241–1251. Association for Computational Linguistics.

Yftah Ziser and Roi Reichart. 2019. Task refinement learning for improved accuracy and stability of unsupervised domain adaptation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5895–5906. Association for Computational Linguistics.

A Additional Generated Examples

We provide additional D-CONs generated by DoCoGen and examples generated by VAE.

(1) D-CONs generated by DoCoGen, for different source-target domain pairs:

(1.1) Original, **DVD**:
the best show on t.v. Somehow Denis Leary and Peter Tolan managed to improve on an already amazing show. Rescue me continues to be the best show out there.

DVD → Airline:
the best service on board. Somehow staff and flight crew were very good and Tolan managed to improve on an already amazing flight. iberia for me continues to be the best airline out there.

DVD → Electronics:
the best wireless router out there. Somehow ipaq and ethernet tolan managed to improve on an already amazing router. ethernet to me continues to be the best wireless out there.

DVD → Kitchen:
the best oven on the market. Somehow a little bit of work and chef Tolan managed to improve on an already amazing oven. this for me continues to be the best oven out there.

(1.2) Original, **Airline**:
Took flight on a one way basis as return dates were subject to change. I am a top tier AA flier and the addition to Air Berlin gives some good connection options. Aircraft was a new Airbus with standard seat pitch and seat comfort in economy was uncomfortable. The IFE is good with each seat having independent controls and a reasonable selection of movies etc. Unfortunately the selection is via a touch screen buried in your seat.

Airline → DVD:
Took this movie on a trial basis as release dates are subject to change. I am a top ten fan and the addition to the bonus disc gives some good insight. The storyline with standard actors and dialogue in the main role was well done. The plot is well done with each actor having independent controls and a reasonable script. Unfortunately the movie is not about a man who is buried in your soul.

Airline → Electronics:
Took this item on a trial basis as release dates are subject to change. I am a top ten speaker and the addition to the base gives some great options. I have an ipod with standard earbuds and the built in microphone was very nice. The sound is clear with each speaker having independent controls and a reasonable bass response. Unfortunately the remote is not a good buy buried in your pocket.

Airline → Kitchen:
Took this item on a regular basis as delivery dates are subject to change. I am a top cook and the addition to the nonstick gives some good heat retention. I use it with standard pans and the clean up in the dishwasher was easy. The pan is very nice and each pan having independent controls and a reasonable size. Unfortunately the lid is too small for a stovetop and buried in your cupboard.

(1.3) Original, **Electronics**:
"How I can test it?" perhaps you don't even need one? build a prototype circuit and test the current arrangement, and see what the noise is like. For a supposedly non-moving reference, see what the ADC is reading like. If it stays steady, and noise is less than 1 LSB, then don't do anything.

Electronics → askubuntu:
"How I can test it?" perhaps you don't even need one? remount your hard drive and test the sda arrangement, and see what the sda is like. For a supposedly non-moving reference, see what the sda is reading like. If it stays in memory, and the sda is less than 1 LSB, then don't do anything.

Electronics → DBA:
"How I can test it?" perhaps you don't even need one? build a table and test the column arrangement, and see what the index is like. For a supposedly non-moving reference, see what the index is reading like. If it stays stale, and the load is less than 1 byte, then don't do anything.

Electronics → Physics:
"How I can test it?" perhaps you don't even need one? heat it up and test the particle arrangement, and see what the velocity is like. For a supposedly non-moving reference, see what the waveform is reading like. If it stays steady, and the speed is less than 1 m/s, then don't do anything.

Electronics → Statistics:
"How I can test it?" perhaps you don't even need one? build a regression model and test the sp arrangement, and see what the plot is like. For a supposedly non-moving reference, see what the plot is reading like. If it stays constant, and the mean is less than 1%, then don't do anything.

(1.4) Original, **Statistics**:
While thinking about methods on how to predict a dependent variable that is ordinal and constrained (0, 1, 2, 3, 5 are the only possible values), I found that an ordered logistic regression is an common approach used in the community and as a strategy, that make sense to me. However, I was trying to understand why another method that came into my mind is a bad idea.

Statistics → Apple: While thinking about methods on how to store music that is ordered by album and song (0, 1, 2, 3, 5 are the only possible ids), I found that an ordered playlist is an option used in the community and as a backup, that make sense to me. However, I was trying to understand why another method that came into my mind is a bad idea.

Statistics → askubuntu: While thinking about methods on how to install a package that is compiled and installed (0, 1, 2, 3, 5 are the only possible repos), I found that an ordered deb package is an common package used in the community and as a result, that make sense to me. However, I was trying to understand why another method that came into my mind is a bad idea.

Statistics → DBA: While thinking about methods on how to store a dependent table that is a single and unique column (0, 1, 2, 3, 5 are the only column), I found that an ordered index is an approach used in the community and as a strategy, that make sense to me. However, I was trying to understand why another method that came into my mind is a bad idea.

Statistics → Electronics: While thinking about methods on how to design a circuit that is programmble and constrained (0, 1, 2, 3, 5 are the only inputs), I found that an ordered voltage is an analog circuit used in the community and as a switch, that make sense to me.However, I was trying to understand why another method that came into my mind is a bad idea.

Statistics → Physics: While thinking about methods on how to construct a particle that is ordered and constrained (0, 1, 2, 3, 5 are the only constraints), I found that an ordered field theory is an approach used in the community and as a general relativist, that make sense to me.However, I was trying to understand why another method that came into my mind is a bad idea.

(2) D-CONs generated by DoCoGen and their masked versions:
(2.1) Original, **Airline**:
One major problem was that the day I flew there was a fuel service strike so I was delayed for over 4 hours. I frequently check my flight status especially when I leave, throughout that day I had checked at least 4 times. I only found out about it when I was at the check in counter. There was no email, no automated phone call, nothing to notify me. I was stuck waiting for hours doing nothing.

Masked text, **Airline → Kitchen**:
One major problem was that the day I [MASK] a [MASK] strike so I was [MASK] for [MASK]. I frequently [MASK] my [MASK] especially when I leave, [MASK] I had [MASK] at least 4 times. I [MASK] out about it when I [MASK]. [MASK], no [MASK] call, [MASK] me. [MASK] for [MASK] doing nothing.

Airline → Kitchen:
One major problem was that the day I got home there was a labor strike so I was left without a coffee maker for the night. I frequently refill my coffee especially when I leave, and I had to replace it at least 4 times. I found out about it when I got home. I sent it back to the store, no one came by to call, and they didn't help me. I sat for over an week doing nothing.

(2.2) Original, **Electronics**:
disappointed with the mouse Though I like the keyboard, I've found two serious problems with the mouse. First, it is very sensitive to movement. I've tried making some adjustments but it is still difficult to control. Finally and more importantly, the mouse burns a set of AA batteries about every 8 days. I only use the mouse about 6 hours a week so I should be getting a lot more battery life. I've bought other Logitech mouse.

Masked text, **Electronics → DVD**:
disappointed with the [MASK] Though I like the [MASK], [MASK] two serious problems with the [MASK]. First, it is very [MASK] to movement. [MASK] making some [MASK] but it is still difficult [MASK]. Finally and more importantly, the [MASK] about every 8 days. I [MASK] the [MASK] about 6 [MASK] week so I should be getting a lot more [MASK] life. I've [MASK] other [MASK].

Electronics → DVD:
disappointed with the workout. Though I like the workout, I have two serious problems with the

workout. First, it is very slow to movement. I'm making some progress but it is still difficult to follow. Finally and more importantly, the workout only goes on about every 8 days. I do the workout about 6 days a week so I should be getting a lot more exercise in my life. I've seen other workouts that aren't slow.

(2.3) Original, **Physics**:
@PaulStonaha: elastic forces cannot provide extra pressure in excess of 10% (balloons pop at such excess pressure), so elastic properties of balloons should not have significant effect.

Masked text, **Physics** → **askubuntu**:
@PaulStonaha: [MASK] cannot provide extra [MASK] in excess of 10% ([MASK] pop [MASK] excess [MASK]), so [MASK] should not have significant effect.

Physics → **askubuntu**:
@PaulStonaha: dpkg cannot provide extra partitions in the in excess of 10% (ubuntu pop installs excess partitions), so redownload should not have significant effect.

(3) D-CONs generated by DoCoGen while providing different orientation vectors:
(3.1) Original, **Airline**:
It was a fantastic flight crew helpful and smiling. All announcements very clear understandable and most important things for me multi - language Turkish English Russian and Kazakh. The way aircraft was new.

Airline → **DVD**, Orientation: "dvd":
It was a fantastic movie. All the main actors were very clear understandable and most important things for me multi - language Russian and Kazakh. The way they talk.

Airline → **DVD**, Orientation: "character":
It was a fantastic movie. All the main character's speech was very clear understandable and most important things for me multi - language Russian and Kazakh. The way it was done was great.

Airline → **DVD**, Orientation: "actor":
It was a fantastic movie. All actors very clear understandable and most important things for me multi - language Russian and Kazakh. The way they act.

Airline → **DVD**, Orientation: "plot":
It was a fantastic movie. All the plots were very clear understandable and most important things for me multi - language Russian and Kazakh. The way it was told was very good.

(4) Examples generated by the VAE baseline, controlling both for the domain and the label:

Airline, **Positive**:
i paid for a trip to hawaii. i was traveling with aeroflot. i was pleasantly surprised. i was pleasantly surprised. the service was excellent and service. i paid for $50 for business class. i am very satisfied with this airline.

Airline, **Positive**:
one of the most memorable movie ever made. i think this movie is a silly comedy, but i was a little silly "attitude of" the "buddy" . "attitude" attitude of the robots, but i was a little silly job of the movie.

Electronics, **Negative**:
not worth the money for my ipod nano. i bought this product for my 3 year old and i am not sure why i am not sure why i am not sure why i am not disappointed.

Kitchen, **Positive**:
broken broken after a broken set of my mother and i needed a gift for my sister. i was skeptical about how to do it. i was able to use it to my dishwasher safe and i was delighted with a silverware. i would recommend it

B Implementation Details

B.1 URLs of Code and Data

- **DoCoGen Repository** - code and datasets: https://github.com/nitaytech/DoCoGen
- **HuggingFace** (Wolf et al., 2020) - code and pretrained weights for the T5 model and tokenizer: https://huggingface.co/
- **SentenceTransformers** (Reimers and Gurevych, 2019) - code and pretrained weights of a LM. We use this LM to extract the embeddings of input examples, and then calculate the cosine similarity between them to match examples in the Oracle-Gen model: https://www.sbert.net/
- **PERL** (Ben-David et al., 2020) - A SOTA unsupervised domain adaptation model: https://github.com/eyalbd2/PERL
- **EDA** (Wei and Zou, 2019) - https://github.com/jasonwei20/eda_nlp
- **VAE** - based on the controllable generation model of Russo et al.

(2020): https://github.com/DS3Lab/control-generate-augment

B.2 Hyperparameters and Setups

Data Preprocessing We 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 examples from the Airline domain, before truncating, we remove the first sentence since it mostly contains details about the flight (like "from JPK to LAX").

DoCoGen Masking: We estimate $P(\mathcal{D}|w)$ for uni-grams, 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 additional 5% of the training examples (in order to add noise between training epochs). We set $\tau = 0.08$ since it resulted in the successfully domain alternation of more than 80% examples. For RM-RR and RM-OV we randomly mask 15% of the examples (the standard ratio for MLM).

Controllable Model: We use $K = 4$ orientation vectors for each unlabeled domain and initialize them with the following representing words for the sentiment dataset: 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, schematic, voltage, circuit; Physics: physics, gravity, particle, quantom; Statistics: stats, regression, logits, variance;

The controllable model is based on a pretrained 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 (Loshchilov and Hutter, 2019) with a learning rate parameter of 5e-5 and a weight decay parameter of 1e-5. For RM-RR and RM-OV we pick the best models based on a MLM loss computed on a held-out set. In the example generation step we use a Beam Search decoding method with a beam size of 4.

VAE As described in the main paper, our VAE implementation is based on Russo et al. (2020). To adjust the model for the purposes of this research, we control the task label and the domain label of each generated review. We train the model on the entire labeled data and unlabeled data that is available from four domains: A, D, E, and K, for a total of 8000 labeled reviews and 104075 unlabeled reviews. We train the VAE for 60 epochs, concatenating sentences with more than 96 tokens, and applying a batch size of 32. The rest of the hyperparameters were set to the values described in Russo et al. (2020).

DA Evaluation Data Augmentation Given a labeled example from the source domain, we generate $K \cdot N = 16$ examples by DoCoGen, where K is the number of orientation vectors of each domain and N is the number of unlabeled domains. We use the generated examples for data augmentation for the task classifiers. For all augmentation models, we apply an augmentation ratio identical to the one used for DoCoGen, yielding augmented training sets of the same size. For NoDA and DANN we duplicate the training set $K \cdot N$ times, thus the number of training steps of all the classifiers is identical. For EDA we use the default hyperparameters.

Task Classifiers All classifiers are based on the T5-encoder architecture equipped with a linear layer, except from PERL which is based on the BERT architecture. We train the classifiers for 5 epochs with a batch size of 64 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.

For the results reported in Tables 3, 4, 8, 9, 5, 10 and 11 we employ a training set that consists of 100 examples and a validation set with 25 examples. To increase the robustness of the results in our small labeled training set setup, we train 25 classifiers, each using a different randomized seed and a randomly sampled training set. We report the average performance of these classifiers on the test set. For the results reported in Figure 2, the valida-

[5] The domain accuracy is measured by a domain-classifier trained on the unlabeled data and that is based on the T5 encoder architecture.

Sentiment Classification				
Domain	Unlabeled	Train	Dev	Test
Airline (A)	39454	1700 (100)	300 (25)	2000
DVDs (D)	34742	1700 (100)	300 (25)	2000
Electronics (E)	13154	1700 (100)	300 (25)	2000
Kitchen (K)	16786	1700 (100)	300 (25)	2000
Books (B)	6001 (0)	1700 (0)	300 (0)	2000
Restaurant (R)	25000 (0)	1700 (0)	300 (0)	2000
Intent Prediction				
Domain	Unlabeled	Train	Dev	Test
Apple (AP)	24752	354 (100)	142 (25)	196
DBA (DB)	25121	311 (100)	138 (25)	199
Electronics (EL)	27192	664 (100)	276 (25)	397
Physics (PH)	25675	142 (100)	68 (25)	78
Statistics (ST)	25743	176 (100)	72 (25)	102
Askubuntu (UB)	26930	1096 (100)	418 (25)	610
DIY (DI)	7383 (0)	0 (0)	0 (25)	180
English (EN)	14734 (0)	0 (0)	0 (0)	189
Gaming (GA)	14050 (0)	0 (0)	0 (0)	117
GIS (GI)	25291 (0)	0 (0)	0 (0)	418
Sci-Fi (SC)	10145 (0)	0 (0)	0 (0)	109
Security (SE)	18302 (0)	0 (0)	0 (0)	109
Travel (TR)	6687 (0)	0 (0)	0 (0)	61
Worldbuilding (WO)	6044 (0)	0 (0)	0 (0)	54

Table 6: Number of available samples in each domain. Numbers in parenthesis represent the amount of samples used for each DA setup.

↗	**A**	**D**	**E**	**K**
A	15.2	37.9	37.3	38.0
D	25.0	16.5	24.0	23.9
E	27.8	26.7	15.7	19.7
K	30.2	28.0	21.1	15.7

Table 7: Percentage of tokens of the original examples that were masked by DoCoGen in the sentiment classification dataset. The left column indicates the source domain and the top row indicates the target domain.

tion set size is 25% of the training size. We train the classifiers on 25 different seeds and partitions for training sizes 25, 50 and 100, and 10 seeds and partitions for sizes 250, 500 and 1000.

B.3 Masking

Table 7 presents the average percentage of masked tokens in the corruption step of DoCoGen (see §4.1), in the sentiment classification dataset. Overall, the average percentage of masked tokens in a single review is 25.2. These statistics emphasize the large gap between original reviews and their D-CONs.

C Ablation Results

C.1 Standard Deviations

Each of the numbers reported in the main result tables of the main paper is the average of 25 repetitions, across seeds and training sets. We hence also report here the standard deviations of these results, which indicate on the stability of the participating models.

The standard deviations for the UDA and ADA setups of sentiment classification are presented in Tables 8 and 9, respectively. F-DoCoGen outperforms all baseline models (NoDA, DANN, EDA, and RM-RR) in 11 of 12 UDA setups and in 6 of 8 ADA setups, demonstrating a lower average standard deviation and an improvement of 22.0% and 27.5% in the UDA and the ADA setups, respectively, over the best performing baseline model. Moreover, DoCoGen without filtering is also superior to all baselines. These results highlight the impact of D-CON generation on model stability in low-resource DA setups.

As noted in the main paper, we also evaluate the complementary effect of DoCoGen and PERL, a SOTA model for UDA. Tables 8 shows that DoCoGen-PERL achieves the lowest average standard deviation, improving PERL by 42%. DoCoGen-PERL is hence the best performing model both in terms of accuracy (see main paper) and in terms of standard deviation (stability).

Tables 10 and 11 report the F1 scores and the standard deviations for the UDA and ADA setups of intent prediction, respectively. As in the case of sentiment classification, F-DoCoGen and DoCoGen are superior to all baselines, achieving lower standard deviation results in the majority of setups. The tables provide additional information regarding the F1 results presented in the main paper (Table 5), reporting F1 scores obtained for each source/target pair experiment.

	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	7.8	6.0	6.8	6.7	5.7	5.4	2.6	4.7	3.0	6.8	4.1	2.9	5.2
DANN	5.4	4.9	5.8	**5.2**	4.5	4.4	3.1	3.4	3.4	2.8	4.4	2.5	4.1
EDA	6.1	5.7	5.8	7.1	6.8	5.4	4.4	4.9	3.5	6.1	4.5	2.9	5.3
RM-RR	6.8	4.9	5.2	5.7	5.1	4.7	3.2	4.3	2.8	5.5	5.1	3.3	4.7
No-OV	8.0	6.8	7.5	6.8	6.1	5.3	3.0	3.1	2.0	5.0	4.8	3.1	5.1
RM-OV	7.6	4.9	5.4	6.7	5.6	4.7	3.8	**2.0**	2.0	7.4	4.8	3.1	4.8
DoCoGen	5.9	4.7	5.1	5.5	4.0	3.5	**1.9**	2.5	2.3	2.2	2.9	1.9	3.5
F-DoCoGen	**4.9**	**4.3**	**4.8**	**5.2**	**3.8**	**3.1**	2.0	2.3	**1.9**	**2.1**	**2.0**	**1.7**	**3.2**
PERL	8.3	5.4	4.6	2.0	6.3	1.2	2.3	2.1	0.7	4.7	4.1	1.4	3.6
DoCoGen-PERL	2.2	0.9	2.7	3.0	1.6	2.1	1.9	1.0	2.8	4.1	1.7	0.9	2.1
Oracle-Gen	1.6	1.2	1.7	1.8	1.0	1.4	0.8	1.2	1.0	1.4	2.9	0.9	1.4

Table 8: Sentiment classification: Standard deviations for each source and target domain pair in the UDA setup. **Bold** numbers mark the best performing T5-based model, and underlined numbers mark the best performing PERL model.

	A → B	A → R	D → B	D → R	E → B	E → R	K → B	K → R	AVG
NoDA	8.0	6.3	3.5	3.7	5.7	4.0	4.1	2.7	4.8
DANN	6.5	6.2	3.3	3.7	3.3	2.2	3.5	4.2	4.1
EDA	**5.9**	4.9	4.1	5.0	5.2	4.3	5.0	3.5	4.7
RM-RR	7.0	4.8	2.9	3.5	5.2	2.9	3.5	2.4	4.0
No-OV	8.2	6.2	2.8	4.0	3.7	**1.6**	4.4	3.1	4.2
RM-OV	7.8	4.9	2.9	4.6	**2.6**	1.9	3.4	3.3	3.9
DoCoGen	7.0	5.7	2.4	3.4	3.2	**1.6**	2.6	2.4	3.5
F-DoCoGen	6.0	**4.0**	**2.0**	**3.3**	3.0	1.7	**1.9**	**1.3**	**2.9**
Oracle-Gen	2.1	2.3	2.0	1.6	1.6	1.8	2.4	1.4	1.9

Table 9: Sentiment classification: Standard deviations for each source and target domain pair in the ADA setup.

	AP → DB	AP → EL	AP → PH	AP → ST	AP → UB	DB → AP	DB → EL	DB → PH	DB → ST	DB → UB
NoDA	77.2 ± 5.3	76.8 ± 5.2	71.4 ± 9.3	74.6 ± 6.7	77.3 ± 4.3	74.6 ± 6.3	74.1 ± 6.1	66.4 ± 11.6	72.6 ± 7.8	73.2 ± 5.3
DANN	77.7 ± 5.0	77.0 ± 4.9	73.2 ± 9.3	74.4 ± 6.5	78.0 ± 4.2	76.1 ± 5.9	74.8 ± 5.5	69.8 ± **9.7**	73.4 ± 7.0	74.7 ± 5.1
EDA	73.7 ± 5.6	72.4 ± 6.1	65.8 ± 9.3	71.1 ± 6.2	74.4 ± 4.8	71.3 ± 6.6	70.6 ± 6.9	63.9 ± 10.2	69.5 ± 7.8	72.0 ± 5.2
RM-RR	77.2 ± 5.7	76.8 ± 4.5	70.9 ± 9.1	74.2 ± 6.3	77.4 ± 4.5	75.1 ± 5.6	75.1 ± 5.3	66.3 ± 10.2	73.5 ± 7.6	73.8 ± 5.1
No-OV	78.4 ± **4.4**	77.7 ± 4.8	72.4 ± 8.4	75.9 ± 5.7	78.3 ± 4.2	75.8 ± 5.7	75.2 ± 4.9	68.3 ± 10.4	73.4 ± **6.4**	74.7 ± 4.6
RM-OV	77.8 ± 5.2	76.6 ± 5.0	69.5 ± 9.2	74.1 ± 6.5	77.2 ± 4.2	75.4 ± 6.1	74.0 ± 6.3	66.4 ± **9.7**	72.6 ± 8.2	74.3 ± 5.2
DoCoGen	**79.2 ± 4.4**	**78.6** ± 3.9	**74.0 ± 7.6**	**76.7** ± 5.3	**78.9 ± 3.7**	**77.1 ± 5.1**	**76.3 ± 4.8**	**70.7 ± 9.7**	**74.9** ± 6.5	**75.9 ± 4.1**
F-DoCoGen	78.6 ± 4.6	78.2 ± **3.8**	73.1 ± **7.6**	76.0 ± **4.7**	78.6 ± **3.7**	77.0 ± 6.3	75.8 ± 5.2	70.2 ± 10.3	74.4 ± 7.3	75.5 ± 4.9
Oracle-Gen	82.6 ± 3.2	81.3 ± 2.5	79.0 ± 5.0	78.4 ± 3.9	82.3 ± 2.4	81.7 ± 3.6	80.2 ± 3.4	76.6 ± 6.5	79.5 ± 4.6	80.2 ± 3.1
	EL → AP	**EL → DB**	**EL → PH**	**EL → ST**	**EL → UB**	**PH → AP**	**PH → DB**	**PH → EL**	**PH → ST**	**PH → UB**
NoDA	72.8 ± 7.2	72.4 ± 7.1	67.6 ± 9.8	71.7 ± 8.0	71.3 ± 7.4	64.5 ± 9.9	67.5 ± 9.3	69.5 ± 7.0	72.8 ± 7.7	61.3 ± 7.7
DANN	74.7 ± 6.2	73.7 ± 6.5	69.6 ± 9.1	72.2 ± 7.0	73.7 ± 6.0	73.1 ± 7.2	72.7 ± 6.5	73.0 ± 5.4	73.9 ± 7.0	70.4 ± 6.2
EDA	70.3 ± 6.6	70.5 ± 6.6	66.1 ± 9.3	70.0 ± **6.2**	69.5 ± 6.4	61.8 ± 6.7	65.5 ± 6.7	67.2 ± 6.2	71.1 ± 6.4	60.2 ± **5.3**
RM-RR	74.2 ± 6.8	73.7 ± 6.9	69.0 ± 9.3	72.0 ± 7.7	72.5 ± 7.0	65.0 ± 8.3	67.8 ± 6.9	70.0 ± 5.9	73.3 ± 6.8	60.8 ± 6.5
No-OV	74.5 ± 6.8	74.1 ± 6.6	69.1 ± 8.6	73.2 ± 7.1	72.5 ± 6.5	68.5 ± 9.4	70.7 ± 8.5	71.5 ± 6.2	75.1 ± 6.6	63.7 ± 7.9
RM-OV	74.5 ± **6.0**	73.5 ± 6.1	67.5 ± 9.2	72.7 ± 7.7	72.6 ± 6.6	67.7 ± 8.0	70.8 ± 6.9	71.9 ± 5.9	75.5 ± 5.8	63.6 ± 6.7
DoCoGen	**76.0 ± 6.0**	**75.9** ± 5.6	**71.0** ± 8.6	**74.6** ± 6.3	**75.0** ± 5.1	75.3 ± 6.6	**74.1** ± 6.1	75.0 ± 4.6	**76.0 ± 5.3**	72.6 ± 6.2
F-DoCoGen	75.5 ± **6.0**	74.8 ± **5.4**	70.6 ± **7.5**	73.6 ± 6.7	74.2 ± **5.0**	**75.5 ± 6.0**	73.8 ± **5.2**	**75.3 ± 4.5**	75.3 ± 5.9	**72.9** ± 5.9
Oracle-Gen	80.4 ± 4.0	78.7 ± 4.5	75.9 ± 6.4	77.9 ± 4.9	78.9 ± 3.2	81.4 ± 3.4	79.8 ± 3.5	79.9 ± 2.4	79.0 ± 3.7	79.2 ± 2.8
	ST → AP	**ST → DB**	**ST → EL**	**ST → PH**	**ST → UB**	**UB → AP**	**UB → DB**	**UB → EL**	**UB → PH**	**UB → ST**
NoDA	70.6 ± 7.1	73.6 ± 5.8	75.0 ± 4.6	70.6 ± 6.8	69.3 ± 6.3	74.6 ± 6.3	73.5 ± 6.0	72.7 ± 6.6	67.0 ± 10.0	72.0 ± 6.7
DANN	74.7 ± 5.1	75.5 ± 4.4	76.5 ± 4.0	72.6 ± 6.8	73.8 ± 4.4	75.4 ± 6.4	74.9 ± 5.7	72.8 ± 6.5	69.3 ± 8.7	71.8 ± 6.6
EDA	68.9 ± 7.8	72.5 ± 5.9	72.4 ± 5.8	68.2 ± 7.3	68.7 ± 6.9	73.2 ± 5.9	72.4 ± 6.2	70.3 ± 6.8	63.4 ± 10.1	69.4 ± 7.3
RM-RR	72.0 ± 7.4	74.6 ± 5.3	75.7 ± 4.5	71.7 ± 6.9	70.3 ± 6.9	76.2 ± 6.0	75.4 ± 5.7	73.8 ± 5.6	66.9 ± **8.6**	72.5 ± 6.9
No-OV	72.7 ± 6.4	74.4 ± 5.7	76.6 ± 4.1	73.5 ± 6.7	70.9 ± 5.8	76.5 ± 5.6	75.2 ± 5.6	74.0 ± 6.3	68.1 ± 9.4	72.7 ± 6.6
RM-OV	71.4 ± 7.2	74.1 ± 6.0	75.2 ± 4.7	70.9 ± 6.7	69.9 ± 6.7	76.6 ± 5.5	75.4 ± **5.3**	74.3 ± 5.6	66.9 ± 9.8	72.5 ± 8.4
DoCoGen	76.4 ± 4.6	**76.4** ± 4.5	78.2 ± 3.6	**75.0** ± 6.2	75.7 ± **3.8**	77.1 ± 5.7	**76.3 ± 5.3**	**75.2 ± 5.5**	**70.9 ± 8.6**	74.5 ± **5.7**
F-DoCoGen	**76.6 ± 4.4**	76.3 ± **3.9**	**78.8 ± 3.1**	73.8 ± **5.2**	**75.8** ± 3.9	**77.2 ± 5.2**	76.2 ± **5.3**	75.0 ± 5.6	69.4 ± 9.8	**74.6** ± 6.7
Oracle-Gen	81.9 ± 3.4	80.5 ± 3.5	80.7 ± 2.8	78.0 ± 4.5	80.7 ± 2.5	83.5 ± 3.4	81.8 ± 3.5	81.4 ± 2.9	79.1 ± 5.1	79.0 ± 3.8

Table 10: Intent prediction: F1 scores and standard deviations for each source and target domain pair in the UDA setup. Each number is calculated across the 5 different task labels, 25 different seeds and randomly sampled training and development sets.

	AP → DI	**AP → EN**	**AP → GA**	**AP → GI**	**AP → SC**	**AP → SE**	**AP → TR**	**AP → WO**
NoDA	74.5 ± 5.4	69.9 ± 6.7	75.5 ± 6.3	76.7 ± 4.6	68.5 ± 6.9	76.9 ± 4.5	79.4 ± 6.5	72.6 ± 9.2
DANN	74.3 ± 5.2	71.9 ± 6.1	76.6 ± 6.2	76.9 ± 4.6	71.0 ± 7.1	77.3 ± 4.5	79.4 ± 7.0	75.4 ± 8.5
EDA	70.6 ± 6.4	66.1 ± 7.0	72.3 ± 6.4	73.2 ± 5.4	62.5 ± 7.4	73.2 ± 5.3	76.4 ± 6.6	68.4 ± 9.8
RM-RR	74.6 ± 5.7	69.9 ± 6.4	75.9 ± 6.2	76.7 ± 4.8	68.4 ± 7.7	76.6 ± 4.9	79.5 ± 6.5	73.2 ± 9.7
No-OV	75.4 ± 4.6	70.9 ± 6.2	76.7 ± 5.8	77.9 ± 4.3	69.7 ± 7.1	78.1 ± 4.2	79.8 ± 6.8	74.3 ± 8.5
RM-OV	74.8 ± 4.9	70.3 ± 6.0	75.9 ± 5.8	76.7 ± 4.3	68.8 ± 7.1	76.8 ± 4.5	78.7 ± 6.2	72.7 ± 8.5
DoCoGen	76.1 ± 4.0	**72.1** ± 5.6	**77.8** ± 5.4	**78.6** ± 4.0	**71.2 ± 6.1**	78.4 ± 3.9	**80.6 ± 6.0**	**77.0** ± 7.5
F-DoCoGen	**76.8 ± 3.3**	**72.1 ± 4.9**	77.7 ± **4.8**	78.3 ± **3.9**	70.8 ± 6.7	**78.5 ± 3.5**	80.0 ± 6.1	75.7 ± **6.4**
Oracle-Gen	79.7 ± 2.8	77.8 ± 3.7	81.8 ± 3.7	81.4 ± 2.6	77.5 ± 4.2	82.1 ± 2.6	83.8 ± 4.9	80.3 ± 5.5
	DB → DI	**DB → EN**	**DB → GA**	**DB → GI**	**DB → SC**	**DB → SE**	**DB → TR**	**DB → WO**
NoDA	71.3 ± 6.5	67.0 ± 7.2	72.2 ± 7.5	73.5 ± 4.6	65.8 ± 8.2	74.2 ± 6.1	73.9 ± 10.2	69.9 ± 10.1
DANN	73.6 ± 5.5	69.8 ± 6.3	74.7 ± 6.6	74.7 ± 4.8	68.5 ± 7.7	75.0 ± 5.5	76.4 ± **7.7**	71.9 ± 8.8
EDA	67.8 ± 7.8	63.9 ± 6.9	69.0 ± 7.5	72.2 ± 4.8	61.1 ± **7.2**	71.4 ± 7.0	70.0 ± 9.8	65.8 ± 9.9
RM-RR	72.0 ± 7.5	67.2 ± 6.5	72.7 ± 7.2	74.7 ± 4.1	65.8 ± 8.3	74.2 ± 5.9	75.0 ± 9.6	68.6 ± 10.6
No-OV	73.0 ± 5.4	68.6 ± 6.4	73.8 ± 6.5	74.6 ± 4.2	67.3 ± 7.8	74.9 ± **5.0**	76.3 ± 8.7	70.9 ± 8.7
RM-OV	72.0 ± 7.8	66.8 ± 7.0	72.0 ± 8.0	74.5 ± 5.2	65.7 ± 8.7	74.5 ± 6.3	73.7 ± 10.6	68.6 ± 10.8
DoCoGen	**74.3 ± 4.9**	**70.3 ± 5.5**	**75.6 ± 6.2**	**75.7 ± 3.8**	**68.8** ± 7.9	**76.1 ± 5.0**	**77.8** ± 8.1	**73.1 ± 8.2**
F-DoCoGen	73.6 ± 6.4	69.8 ± 6.4	75.3 ± 6.7	75.2 ± 4.7	68.0 ± 7.5	75.9 ± 5.4	77.0 ± 8.4	71.6 ± 10.1
Oracle-Gen	78.9 ± 3.1	75.5 ± 4.1	80.0 ± 4.2	79.7 ± 3.4	76.0 ± 5.2	80.6 ± 3.5	83.6 ± 5.2	80.0 ± 5.4
	EL → DI	**EL → EN**	**EL → GA**	**EL → GI**	**EL → SC**	**EL → SE**	**EL → TR**	**EL → WO**
NoDA	72.5 ± 6.3	67.2 ± 7.9	69.5 ± 9.4	71.7 ± 7.0	66.3 ± 8.9	74.0 ± 6.7	74.1 ± 10.8	70.7 ± 10.7
DANN	73.2 ± 6.3	69.2 ± 6.6	72.3 ± 8.2	73.5 ± 5.8	68.1 ± 8.3	75.3 ± 5.7	76.4 ± 9.4	71.9 ± 9.9
EDA	71.1 ± 6.2	64.0 ± 6.6	67.7 ± 8.4	70.3 ± 6.1	62.5 ± 7.6	72.1 ± 5.6	71.2 ± 8.2	70.4 ± **8.7**
RM-RR	73.7 ± 5.9	67.5 ± 7.8	70.6 ± 8.8	73.1 ± 6.2	66.7 ± 8.1	76.0 ± 5.8	74.9 ± 9.1	70.9 ± 9.5
No-OV	74.4 ± 5.1	69.0 ± 6.6	71.9 ± 8.7	73.2 ± 5.9	68.2 ± 8.2	75.6 ± 5.9	76.8 ± 9.2	72.1 ± 10.2
RM-OV	74.1 ± 5.4	68.1 ± 7.3	71.9 ± 7.8	73.2 ± 5.6	67.3 ± 8.8	76.7 ± 5.7	75.6 ± 8.4	71.3 ± 8.9
DoCoGen	**75.1** ± 5.0	**70.7 ± 5.7**	**74.0 ± 7.3**	**74.7** ± 5.3	**69.1** ± 7.4	77.2 ± 5.5	**78.9** ± 7.6	**73.9** ± 9.2
F-DoCoGen	74.6 ± **4.7**	69.8 ± 5.9	72.6 ± 7.6	73.9 ± **5.0**	67.8 ± **7.3**	**77.4 ± 4.6**	77.1 ± **6.8**	72.1 ± 9.5
Oracle-Gen	77.8 ± 3.5	76.5 ± 4.5	79.2 ± 5.0	78.7 ± 3.6	76.4 ± 5.8	80.3 ± 3.7	81.5 ± 5.3	80.1 ± 5.8
	PH → DI	**PH → EN**	**PH → GA**	**PH → GI**	**PH → SC**	**PH → SE**	**PH → TR**	**PH → WO**
NoDA	66.5 ± 8.9	65.4 ± 5.7	64.8 ± 8.3	66.1 ± 8.7	70.2 ± 8.8	68.7 ± 8.6	64.0 ± 9.3	70.2 ± 9.5
DANN	71.7 ± 6.7	69.0 ± 5.0	71.5 ± 7.8	71.4 ± 6.0	73.1 ± 7.3	75.5 ± 5.3	71.0 ± 8.7	72.9 ± 7.7
EDA	64.8 ± 7.7	61.9 ± 5.1	61.8 ± **6.0**	65.8 ± 6.1	66.3 ± 7.4	66.0 ± 6.5	61.8 ± **7.5**	68.7 ± 7.7
RM-RR	66.9 ± 7.4	65.7 ± 5.8	65.4 ± 6.9	67.2 ± 6.8	71.3 ± 8.2	69.9 ± 7.1	64.6 ± 9.7	69.4 ± 8.9
No-OV	69.9 ± 7.5	69.0 ± 4.7	68.3 ± 8.4	68.8 ± 7.9	72.1 ± 8.1	72.1 ± 7.1	68.7 ± 9.2	73.5 ± 7.0
RM-OV	70.0 ± 6.8	67.7 ± 5.3	67.7 ± 7.3	69.5 ± 6.3	73.4 ± 6.9	71.7 ± 7.4	66.9 ± 9.1	73.7 ± 7.9
DoCoGen	**73.7** ± 5.6	71.4 ± **4.1**	74.2 ± 7.5	**73.0** ± 5.8	**74.6** ± 6.1	76.7 ± 5.0	74.0 ± **7.5**	75.1 ± **6.5**
F-DoCoGen	**73.7 ± 5.1**	**72.6** ± 4.4	**75.3** ± 6.4	72.6 ± **5.5**	74.5 ± **5.5**	**77.5 ± 4.9**	**74.6** ± 7.7	**75.9** ± 7.7
Oracle-Gen	78.0 ± 2.8	76.7 ± 3.8	80.9 ± 3.9	79.4 ± 2.8	78.6 ± 4.1	81.6 ± 2.7	81.4 ± 5.1	80.9 ± 5.0
	ST → DI	**ST → EN**	**ST → GA**	**ST → GI**	**ST → SC**	**ST → SE**	**ST → TR**	**ST → WO**
NoDA	70.1 ± 6.8	68.5 ± 5.2	66.5 ± 6.8	73.8 ± 5.1	67.2 ± 7.8	74.5 ± 4.5	68.8 ± 10.2	70.7 ± 7.4
DANN	73.5 ± 5.3	70.2 ± 5.5	70.6 ± **4.9**	74.9 ± 4.1	69.1 ± 6.7	76.7 ± 3.8	74.4 ± 7.4	73.1 ± 6.8
EDA	69.6 ± 7.0	66.9 ± 4.6	66.4 ± 8.0	72.8 ± 5.6	63.8 ± 6.2	72.1 ± 5.8	70.6 ± 10.3	69.4 ± 7.9
RM-RR	71.7 ± 6.7	69.5 ± 5.1	68.0 ± 7.4	74.7 ± 4.9	68.5 ± 6.8	75.2 ± 5.1	70.6 ± 10.1	71.4 ± 7.3
No-OV	72.4 ± 5.6	71.0 ± 5.2	67.9 ± 6.4	74.5 ± 5.1	69.9 ± 7.2	75.4 ± 4.4	73.2 ± 8.1	73.6 ± 7.2
RM-OV	70.9 ± 6.8	69.5 ± 4.5	68.0 ± 7.4	74.5 ± 5.5	69.3 ± 7.0	75.4 ± 4.8	69.8 ± 10.2	73.1 ± 7.0
DoCoGen	75.4 ± 4.3	72.5 ± 4.7	71.9 ± 6.0	76.2 ± **3.8**	70.3 ± 5.8	77.5 ± 4.1	**77.9 ± 6.2**	**75.0** ± 6.9
F-DoCoGen	**76.0 ± 3.5**	**72.6 ± 4.3**	**72.8** ± 5.1	**76.6 ± 3.8**	**70.8 ± 5.2**	**77.9 ± 2.9**	77.3 ± 6.5	74.4 ± **6.7**
Oracle-Gen	78.3 ± 2.7	75.0 ± 3.0	79.9 ± 3.3	80.1 ± 2.5	75.8 ± 4.9	80.1 ± 2.9	83.5 ± 5.4	81.0 ± 4.3
	UB → DI	**UB → EN**	**UB → GA**	**UB → GI**	**UB → SC**	**UB → SE**	**UB → TR**	**UB → WO**
NoDA	71.7 ± 7.5	66.6 ± 6.9	73.6 ± 7.3	72.9 ± 5.4	67.6 ± 7.6	74.2 ± 6.0	73.9 ± 9.4	68.4 ± 9.1
DANN	72.1 ± 6.6	69.1 ± 7.2	74.1 ± 6.5	73.5 ± 5.3	**71.4 ± 7.4**	75.0 ± 5.5	75.5 ± **8.1**	71.3 ± **7.9**
EDA	68.4 ± 8.3	63.5 ± 6.7	72.0 ± 7.3	71.7 ± 5.6	62.5 ± 7.7	71.4 ± 6.5	69.9 ± 10.7	64.6 ± 9.8
RM-RR	73.1 ± 7.1	66.7 ± 6.6	74.2 ± 7.4	74.5 ± 5.0	68.7 ± 7.9	74.8 ± 5.4	73.8 ± 10.6	68.8 ± 10.0
No-OV	72.8 ± 7.0	67.9 ± 6.4	74.5 ± 6.7	73.9 ± 5.1	68.7 ± 8.0	74.8 ± 5.7	75.3 ± 9.0	70.4 ± 9.5
RM-OV	73.4 ± 6.7	67.5 ± 6.2	74.7 ± 6.9	75.0 ± **4.7**	68.2 ± **7.4**	75.2 ± **5.0**	74.9 ± 11.1	69.5 ± 9.8
DoCoGen	74.6 ± 6.0	**70.4 ± 6.1**	75.8 ± **6.1**	**75.6 ± 4.7**	70.7 ± 7.5	**76.3** ± 5.4	77.4 ± **8.1**	72.2 ± 8.8
F-DoCoGen	**75.1 ± 5.8**	70.3 ± **6.1**	**76.3** ± 6.2	75.4 ± 4.9	70.0 ± 7.8	76.0 ± 5.6	**77.9** ± 8.9	**72.3** ± 8.5
Oracle-Gen	79.5 ± 3.5	77.2 ± 3.5	81.9 ± 3.5	80.7 ± 3.0	78.8 ± 4.2	81.0 ± 3.4	84.2 ± 5.0	80.4 ± 4.8

Table 11: Intent prediction: F1 scores and standard deviations for each source and target domain pair in the ADA setup. Each number is calculated across the 5 different task labels, 25 different seeds and randomly sampled training and development sets