DialAug: Mixing up Dialogue Contexts in Contrastive Learning for Robust Conversational Modeling

Lahari Poddar **Peiyao Wang** **Julia Reinspach**
Amazon
{poddarl, peiyaow, reinspac}@amazon.com

Abstract

Retrieval-based conversational systems learn to rank response candidates for a given dialogue context by computing the similarity between their vector representations. However, training on a single textual form of the multi-turn context limits the ability of a model to learn representations that generalize to natural perturbations seen during inference. In this paper we propose a framework that incorporates augmented versions of a dialogue context into the learning objective. We utilize contrastive learning as an auxiliary objective to learn robust dialogue context representations that are invariant to perturbations injected through the augmentation method. We experiment with four benchmark dialogue datasets and demonstrate that our framework combines well with existing augmentation methods and can significantly improve over baseline BERT-based ranking architectures. Furthermore, we propose a novel data augmentation method, ConMix, that adds token level perturbations through stochastic mixing of tokens from other contexts in the batch. We show that our proposed augmentation method outperforms previous data augmentation approaches, and provides dialogue representations that are more robust to common perturbations seen during inference.

1 Introduction

Conversational systems have gained immense research attention in the past few years due to their practical applications in building intelligent digital assistants. In order to converse with humans in natural language, a conversational system needs to produce meaningful and contextual responses at every turn of a dialogue. This is often accomplished by a ranking model, the goal of which is to select the most appropriate response among a set of curated candidate responses (Lu et al., 2019; Mehri et al., 2019; Henderson et al., 2019; Xu et al., 2021; Gu et al., 2020; Whang et al., 2020; Han et al., 2021). For practical applications a Bi-encoder model architecture is often adopted, due to its computational efficiency (Humeau et al., 2019; Reimers and Gurevych, 2019; Wu et al., 2020a; Henderson et al., 2019). In this approach, the dialogue context and candidate responses are encoded into latent vectors separately, and the ranking scores are computed based on the similarity between these vectors.

Learning effective vector representations of dialogue contexts is a challenging task. Since most dialogue datasets consist of free-text multi-turn interactions between humans, the *exact* same context-response pair is likely to be seen only *once* in the whole training set. However, during inference, the same response could be appropriate for various different forms of contexts. For example, in the customer service domain, a response such as “I can issue a refund for the damaged item” could be appropriate for many contexts that fall into the general theme of a customer having received a damaged item. Such contexts may differ from one another due to variations in customer language, the particular item details, or the type of damage etc. Hence, a response ranking model needs to learn representations that are robust to syntactic and fine-grained semantic variations in the dialogue context.

In order to learn representations with improved generalization capabilities, data augmentation has become ubiquitous in computer vision (Shorten and Khoshgoftaar, 2019). Recent research (Shen et al., 2020; Feng et al., 2021; Longpre et al., 2020) has also reported success in leveraging augmentations for NLP tasks. An effective method of incorporating data augmentation for better representation learning is through a contrastive learning framework (Chen et al., 2020; Wu et al., 2020b; Gao et al., 2021; Fang and Xie, 2020; Xie et al., 2020; Fabbri et al., 2021; Wei et al., 2021), where the objective is to maximize similarity between encoded representations of an input and its augmented version. While contrastive learning with data augmentations has shown promising results in several NLP tasks, to the best of our knowledge the potential of such approaches for conversational modeling has not yet been explored.

In this work we propose a multi-objective model architecture, DialAug, for learning robust dialogue response ranking. The proposed architecture leverages the power of text data augmentations in combination with contrastive learning. During training, the model learns to predict the same response for both the original dialogue context and for its augmented version, thus making it agnostic to variations in the context. To capture the notion of coherence and semantic relevance of a dialogue, we introduce an auxiliary contrastive objective that learns the similarity between different views of a dialogue context, in contrast to views of contexts of other dialogues.

We further propose a novel data augmentation method for **Con**text **Mix**ing, namely ConMix, that adds token level perturbations to the dialogue context. The aim of introducing the perturbations is to simulate different variations of a multi-turn context. ConMix stochastically replaces some of the input tokens in a dialogue context with tokens from another randomly sampled context in the training batch. The benefits of this method are twofold. First, we are creating a perturbed version of the original context that will help learn generalizable representations. Second, we are also generating hard negatives for other responses and contexts in the batch, due to the word overlap infused through stochastic mixing from other context in the same training batch. To summarize, in this paper we make the following major contributions:

- We propose a multi-objective model architecture, DialAug, for dialogue response ranking that uses a ranking objective and a contrastive learning objective. The proposed architecture is modular and can be effectively combined with many data augmentation techniques.

- We propose a novel data augmentation technique, ConMix, that stochastically adds token-level perturbations to dialogue contexts during training, leading to better performance and robustness of the learned model as compared to baseline data augmentation methods.

- We conduct an extensive set of evaluations on four large-scale publicly available dialogue datasets, and demonstrate the proposed approach outperforms strong baselines and is effective in learning robust representations.

2 Related work

We review two closely related research areas: data augmentation techniques for text data, and contrastive learning.

Data Augmentation for Text : Data augmentation has been widely used for computer vision tasks, in order to increase the size of a labeled dataset, and to improve robustness of the model to input noise. Typical image augmentations include cropping, flipping, rotating, resizing, applying color distortions, and Gaussian blurring (Shorten and Khoshgoftaar, 2019; Chen et al., 2020). Equivalent simple augmentation techniques have been proposed and explored for text data tasks, e.g., word deletions and permutations, and have been shown to improve the model's robustness and performance (Shorten and Khoshgoftaar, 2019). There has also been some active research into semantic augmentation techniques, such as back-translation, synonym replacement, or generative models (Shorten and Khoshgoftaar, 2019; Wu et al., 2020b; Xie et al., 2020; Kumar et al., 2019; Fang and Xie, 2020). However, these are comparatively complex to implement, and rely on external knowledge (i.e., synonym lists) or additional models, making them only suitable for tasks where appropriate models or knowledge exists. In this work we only consider automatic data augmentation techniques, i.e., techniques that do not require external knowledge or additional models, and can be easily implemented for any language or task.

Contrastive Learning : Contrastive learning has been shown to be a powerful representation learning technique for both vision and text data tasks (Chen et al., 2020; Khosla et al., 2020; Wu et al., 2020b; Giorgi et al., 2020; Gunel et al., 2020). It essentially aims to learn a better representation of the input by maximizing agreement between two similar data points. These data points can be either augmented versions of the same input in self-supervised learning (Chen et al., 2020; Giorgi et al., 2020; Wu et al., 2020b), or from the same class label in supervised learning (Gunel et al., 2020; Ma et al., 2021; Khosla et al., 2020).

Contrastive learning has been explored for both pretraining and finetuning tasks in NLP. For example, (Wu et al., 2020b; Giorgi et al., 2020; Fang and

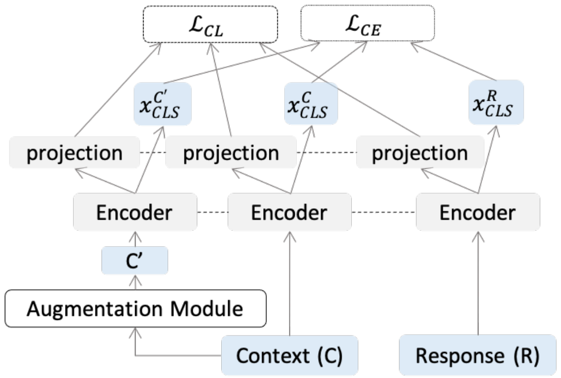


Figure 1: DialAug model architecture. Context C and Response R are the model inputs, and the output is a similarity score. The augmented context C' and projection network are used only during training. The encoder networks share weights, as well as the projection layers.

Xie, 2020) use contrastive learning to pretrain large-scale transformer encoders for sentence representations, while other researchers focus on more task-specific finetuning settings, such as summarization (Fabbri et al., 2021), text classification (Wei et al., 2021), textual similarity (Gao et al., 2021), and user satisfaction prediction (Kachuee et al., 2021). Recent work (Ma et al., 2021) has demonstrated success in adopting contrastive finetuning for neural rankers in the QA domain, however, the authors do not leverage data augmentations. Our work is more similar to CLEAR (Wu et al., 2020b) which uses contrastive learning with text data augmentations for pretraining language models. However, we focus on the finetuning stage of dialogue response ranking and leverage augmentations for dialogue contexts in the contrastive learning objective. We use deletion and reordering based augmentations proposed in their work as baselines for ConMix.

3 Approach

Consider a batch with B inputs $\{C_i, R_i\}_{i=\{1,2,\cdots,B\}}$, where C_i is a dialogue context and R_i is the corresponding response. Given the dialogue context C_i, objective of the model is to predict the most likely response R_i among a set of candidate responses $\{R_1, R_2, \cdots R_m\}$.

3.1 DialAug Model Architecture

We build upon the widely used Bi-encoder model architecture (Humeau et al., 2019; Reimers and Gurevych, 2019; Wu et al., 2020a; Henderson et al., 2019), which is efficient for real world use cases, due to its fast training and inference speed. Figure 1 shows the proposed model architecture.

Our architecture consists of an augmentation module that creates an additional view C'_i of the dialogue context C_i, through certain transformations (described later). We first encode the context C_i, the augmented context C'_i, and the response R_i to latent vectors, using a shared encoder. We leverage pre-trained language models and use BERT (Devlin et al., 2019) as the encoder block. The input sequence to the BERT encoder is represented as

$$C_i = [CLS, w_1 \cdots, EOT, w_j, \cdots, w_{n-1}, EOT] \tag{1}$$

where n is the number of words in the context or response, CLS is a special token marking the beginning of the input sequence, and an additional end-of-turn EOT token marks the end of turns in the dialogue context. We feed these sequences to the BERT encoder and obtain latent vector representations for the input context, the augmented context, and the response.

3.1.1 Main Task Loss

Dialogue contexts consist of multiple turns, with a lot of information that might be redundant for predicting the next response. We argue that such lengthy contexts can usually accommodate small word level variations without changing the overall theme or topic of the conversation and the next response. Therefore, we consider the augmented version of a context C'_i to be label-invariant.

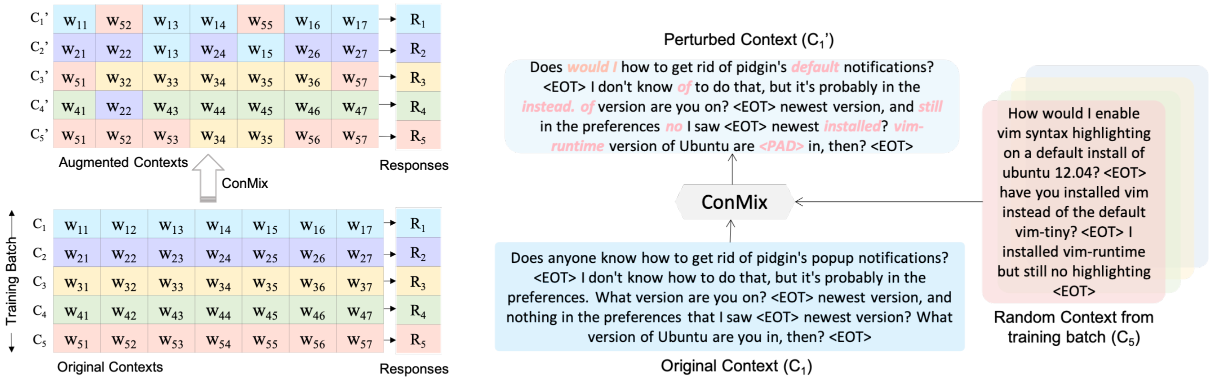
This allows the model to learn that R_i is the next response for both the original context C_i and its augmented version C'_i. Introducing these (C'_i, R_i) pairs in the main task loss of response ranking essentially doubles the number of training data points seen by the model in each epoch. More importantly, this forces the model to learn robust representations of the lengthy dialogue contexts, in order to rank the response R_i over other m candidate responses, for two different views of it.

In order to obtain an aggregated vector representation of the sequences, we use the latent vector representation of the CLS token. The score of a candidate response R_i, for a context, is computed using dot-product of their vector representations

$$score(C_i, R_i) = x^{C_i}_{CLS} \cdot x^{R_i}_{CLS} \tag{2}$$

$$score(C'_i, R_i) = x^{C'_i}_{CLS} \cdot x^{R_i}_{CLS} \tag{3}$$

where $x^{C_i}_{CLS}, x^{C'_i}_{CLS}, x^{R_i}_{CLS}$ denote the representations from the CLS token of context, the aug-



C_1'	w_{11}	w_{52}	w_{13}	w_{14}	w_{55}	w_{16}	w_{17}	R_1
C_2'	w_{21}	w_{22}	w_{13}	w_{24}	w_{15}	w_{26}	w_{27}	R_2
C_3'	w_{51}	w_{32}	w_{33}	w_{34}	w_{35}	w_{36}	w_{57}	R_3
C_4'	w_{41}	w_{22}	w_{43}	w_{44}	w_{45}	w_{46}	w_{47}	R_4
C_5'	w_{51}	w_{52}	w_{53}	w_{34}	w_{35}	w_{56}	w_{57}	R_5

C_1	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}	w_{16}	w_{17}	R_1
C_2	w_{21}	w_{22}	w_{23}	w_{24}	w_{25}	w_{26}	w_{27}	R_2
C_3	w_{31}	w_{32}	w_{33}	w_{34}	w_{35}	w_{36}	w_{37}	R_3
C_4	w_{41}	w_{42}	w_{43}	w_{44}	w_{45}	w_{46}	w_{47}	R_4
C_5	w_{51}	w_{52}	w_{53}	w_{54}	w_{55}	w_{56}	w_{57}	R_5

(a) Mixing context tokens in batch

Perturbed Context (C_1')

Does *would I* how to get rid of pidgin's *default* notifications? <EOT> I don't know *of* to do that, but it's probably in the *instead. of* version are you on? <EOT> newest version, and *still* in the preferences *no* I saw <EOT> newest *installed*? *vim-runtime* version of Ubuntu are *<PAD>* in, then? <EOT>

Does anyone know how to get rid of pidgin's popup notifications? <EOT> I don't know how to do that, but it's probably in the preferences. What version are you on? <EOT> newest version, and nothing in the preferences that I saw <EOT> newest version? What version of Ubuntu are you in, then? <EOT>

Original Context (C_1)

How would I enable vim syntax highlighting on a default install of ubuntu 12.04? <EOT> have you installed vim instead of the default vim-tiny? <EOT> I installed vim-runtime but still no highlighting <EOT>

Random Context from training batch (C_5)

(b) Example of generated perturbed context through augmentation

Figure 2: Illustration of the ConMix data augmentation. C_1 is a context and R_1 is its corresponding response. C_1' is an augmented version of C_1, which retains most of the tokens from C_1 (blue), and has few tokens replaced with tokens from a random context C_5 (orange). R_1 is still considered the most appropriate response to C_1'.

mented context and the response, respectively. We optimize a cross-entropy loss ($\mathcal{L}_{CE}$) to achieve our main goal of scoring the next response in the dialogue higher than a set of candidate responses. During training, we consider the other responses in a batch as negatives for a given context.

3.2 Contrastive Learning

We introduce a contrastive learning objective as an auxiliary task during training. In particular, through the contrastive learning objective, we learn similarities between the vector representations of the original context C_i and the augmented context C_i'. In addition to C_i and C_i', the response R_i is also a part of the same dialogue, and hence we include the response candidates into our contrastive loss.

Following (Chen et al., 2020), we apply a projection network $g(\cdot)$ to transform the representations to a space where the contrastive loss will be applied. We use a simple 2-layer feed-forward network with ReLU non-linearity. The contrastive loss $\mathcal{L}_{CL}$ is optimized to maximize the similarity between span representations of C, C' and R.

We adopt a generalized version of the NT-Xent loss (Chen et al., 2020) that can accept multiple positives. For the i^{th} training instance, the positive pairs are given by $\{(z_{C_i}, z_{C_i'}), (z_{C_i}, z_{R_i}), (z_{C_i'}, z_{R_i})\}$. For each such positive pairs (p_i, p_i^+), the contrastive loss term is represented as

$$\ell_{p_i,p_i^+} = -\log \frac{exp(z_{p_i} \cdot z_{p_i^+})/\tau}{\sum_{k=0}^{B} \mathbb{1}_{k \neq i} \sum_{q \in S} exp(z_{p_i} \cdot z_{q_k})/\tau} \tag{4}$$

where τ denotes the temperature in the loss, $\mathbb{1}_{k \neq i}$ is an indicator function, B is the batch size and $S = \{C, C', R\}$ are the sequences in the batch. The total contrastive loss $\mathcal{L}_{CL}$ within a batch is computed over all such positive pairs.

The overall loss is a weighted summation of the cross-entropy and the contrastive loss,

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{CL} \tag{5}$$

where λ is a weight coefficient for the auxiliary loss. We empirically set this value to 0.5 for all our experiments. A careful reader might observe that we introduce additional parameters in the skeleton Bi-encoder architecture through the projection network, however, they are only used during training and discarded afterwards. During inference, the model has a comparable number of parameters and speed as a Bi-encoder.

3.3 ConMix Data Augmentation

We design a novel data augmentation method, ConMix, to generate the augmented view C_i' for a given context C_i. ConMix creates augmentations through dynamic mixing of words from other contexts in the batch. In particular, for each C_i it selects a random context C_j from the training batch and replaces random words of C_i with words in the same positions from C_j, to generate an augmented version (C_i'). Figure 2 shows an illustrative example of the mixing process. This introduces perturbations to the original context and stochastically creates variations which the model learns to recognize as similar, and ranks the same response at the top among other candidates. With the batch mixing strategy the augmented context (C_i') also serves as a hard negative. This is because the augmented version C'_i has significant word overlap with C_j, the random context from where the replacement tokens were chosen. Thus creating harder negative pairs <C'_i, R_j> and <C'_i, C_j> in the main task loss and the contrastive loss, respectively.

We adapt the Bernoulli MixUp approach (Beckham et al., 2019) for mixing tokens of dialogue contexts. In C'_i, we wish to retain the *majority* of tokens from the original context C_i, and replace the rest with tokens from a random context C_j. We first sample a binary mask $m \in \{0, 1\}^n$, where n is the number of tokens in a context sequence.

$$C'_i = m \circ C_i + (1 - m) \circ C_j, \text{ where } i \neq j \quad (6)$$

C'_i is the augmented view of context C_i, and C_j is a randomly selected context from the same batch, $\circ$ denotes the Hadamard product. The binary mask m is sampled from a Bernoulli(λ_{mix}) distribution where $\lambda_{mix} \in (0.5, 1]$ is the mixing coefficient. The proportion of replaced tokens is controlled by λ_{mix}. Intuitively, we should use a coefficient that retains most of the words from the original context, to ensure that the augmented context C'_i is label invariant, i.e., can have the same next response R_i, and is more similar to C_i than to C_j. In order to preserve the higher-level dialogue structure, we retain the end-of-turn (EOT) markers in the context while generating the binary mask.

The augmentations in our architecture are stochastically generated during each epoch. Therefore, for a context C_i, the augmented view C'_i might be different across epochs, depending on the random selection of the mixing context C_j and replaced token positions within C_i. This enables the model to see many variations of the same context and learn to generalize across these representations.

4 Experiments

4.1 Datasets

We finetune and evaluate our response ranking models on the following four public task-oriented dialogue datasets:

1. Ubuntu V2: The Ubuntu V2 corpus (Lowe et al., 2015) consists of conversations extracted from Ubuntu chat logs, where people seek technical support for various Ubuntu-related problems from the community. We use a public repository[1] to generate the train/dev/test examples.

	Ubuntu v2	DSTC7		Taskmaster
		Ubuntu	Advising	
Train examples	1M	100k	100k	192,821
Dev examples	19,560	5k	500	10,715
Test examples	18,920	1k	500	10,717
Eval candidates	10	100	100	51

Table 1: Statistics of the datasets: Ubuntu V2, Ubuntu DSTC7, Advising DSTC7, and Taskmaster.

2. Advising DSTC7: The Advising dataset from DSTC7 subtask 1 (Gunasekara et al., 2019) contains dialogues in which university students seek advise on classes to take. The dataset was built upon expanding 815 original conversations by paraphrasing. This dataset additionally contains profile information for students, which we do not include in our model to be consistent with other datasets.

3. Ubuntu DSTC7: The Ubuntu DSTC7 dataset is similar to the Ubuntu V2 corpus, but it was further disentangled and annotated from the original chat logs data (Kummerfeld et al., 2018). We evaluate our model on the subtask 1 of the DSTC7 challenge, the goal of which is to select the most appropriate response from 100 candidates.

4. Taskmaster: This dataset consists of written dialogues in the movie ticketing domain (Byrne et al., 2019). We split the dialogues into train/dev/test sets, and treat each system turn and its corresponding dialogue context as a positive pair. For evaluation, we randomly sample 50 negative responses per context from all available system turns.

The dataset statistics are summarized in Table 1. For each dataset, we calculate the 95th percentile of its context and response lengths, and use these values as the maximum sequence length in the corresponding encoders. We use a batch size of 20 for Taskmaster and 32 for the other three datasets.

4.2 Implementation Details

We implement our models using the Pytorch deep learning framework and the HuggingFace transformer library (Wolf et al., 2020). For implementation of the contrastive loss we use the Pytorch metric learning library (Musgrave et al., 2020). We set the mixing coefficient (λ_{mix}) in ConMix to 0.7, i.e., 30% of the tokens are replaced. We use `bert-base-uncased` as our pre-trained encoder, and train all our models in an end-to-end manner with Adam optimizer (Kingma and Ba, 2015) for fine-tuning.

[1] https://github.com/rkadlec/ubuntu-ranking-dataset-creator

4.3 Baseline Augmentations

We explore and evaluate the following augmentation methods as baselines to compare with ConMix:
1. Subsequence sampling: Similar to cropping (Chen et al., 2020) for images and span sampling (Giorgi et al., 2020) for sentences, we explore a subsequence sampling augmentation for dialogues. We create augmentations by randomly truncating the initial turns of a given context. We argue that a response is more closely related to later turns in the context compared to earlier ones, especially in task-oriented dialogues. Hence such a strategy can highly preserve the label from the original context.
2. Word deletion: We implement the word deletion augmentation and hyperparameters as described in (Wu et al., 2020b). Following (Wu et al., 2020b), we randomly select 70% [2] of the words in the dialogue history and replace them with the special token $[DEL]$. We merge consecutive $[DEL]$ tokens into one.
3. Word reordering: We randomly sample several pairs of words in a dialogue context, and switch them pairwise. We swap 30% of the words similar to our proposed ConMix. In contrast to ConMix, this method only mixes words within a single dialogue context.
4. Word replacement: We randomly replace 30% of the words in a context with random words. In contrast to ConMix, this method replaces context words with words from the full vocabulary, and not only with words from the same training batch.

Similar to ConMix we protect the special token EOT from being replaced in all baseline augmentations to preserve the dialogue structure.

5 Results and Discussion

We use Recall@1 and MRR as evaluation metrics and report numbers after averaging over 3 runs.

5.1 Performance on Response Ranking

We first demonstrate our proposed model architecture's compatibility and effectiveness with ConMix, along with other baseline data augmentations. For each augmentation method, we conduct an ablation study to separately understand the effects from data augmentation, and the benefits obtained from the addition of contrastive learning. For a fair baseline comparison we include Bi-encoder (Humeau et al., 2019), which has a comparable number of parameters and architecture. Larger model architectures such as Poly-encoder (Humeau et al., 2019) or Cross-encoder (Wolf et al., 2019) are orthogonal to our approach, and can potentially be adopted as backbone architecture for our model. We leave those explorations for future work.

Results on four ranking datasets for all model variants are presented in Table 2. We observe that our proposed DialAug architecture significantly outperforms the baselines across all datasets. Specifically, our model with proposed ConMix augmentation and contrastive loss achieves an absolute gain of 0.8%, 1.9%, 1.0% and 2.3% for Recall@1 metric over Bi-encoder, on the four datasets respectively. This shows that textual variations injected in the input sequences through augmentations result in representations that generalize better to the unseen test set.

Second, we note that our proposed augmentation method, ConMix, consistently outperforms the baseline augmentations in all datasets by a fair margin, except for Ubuntu DSTC7. We find that the word reordering augmentation, which shuffles words within a context, is not as effective as the other augmentations. In this method, words are neither introduced nor removed from the context, and the model learns from the same bag-of-words as the original context. On the other hand, through deletion augmentation words get omitted from the context, and the model needs to learn to predict the response while some words are missing. ConMix takes this a step further, and not only removes some of the words from the context, but also replaces them with other random words. This forces the model to learn the task in a much harder setting with observing many variations of the same context over the epochs. As hypothesized ConMix outperforms the global word replacement method due to the added advantage of strategic in-batch mixing, infusing word overlaps in a controlled manner and supplementing harder negatives.

Finally, we note that contrastive learning (rows with + CL) helps boost performance further, compared to corresponding model versions without the additional objective. This indicates the effectiveness of learning to contrast partial views of a dialogue for better representation learning of the context. Moreover, we see that for relatively smaller sized dataset from the DSTC7 challenge, contrastive learning acts as an effective regularizer

[2] We also conducted experiments with word deletion rate of 30% similar to ConMix but it underperformed the variant with recommended 70% deletion rate

Models	Ubuntu V2		Advising DSTC7		Ubuntu DSTC7		Taskmaster	
	R@1/10	MRR	R@1/100	MRR	R@1/100	MRR	R@1/50	MRR
Bi-Encoder	82.8±.3	89.5±.2	21.1±.4	33.3±.1	56.7±.7	66.0±.3	87.8±.2	89.6±.2
DialAug + Subsequence	83.0±.0	89.7±.0	21.6±.3	34.2±.1	57.1±.8	66.7±.7	86.9±.2	89.1±.1
DialAug + Subsequence + CL	82.9±.1	89.6±.0	20.6±.3	33.1±.3	56.9±.1	66.7±.2	87.6±.2	89.5±.1
DialAug + Deletion	83.2±.1	89.8±.1	21.5±.1	34.2±.6	57.4±.6	67.2±.5	88.2±.3	90.1±.2
DialAug + Deletion + CL	83.3±.1	89.8±.1	21.7±.1	34.9±.3	**58.1**±.4	**67.8**±.6	88.5±.2	90.2±.1
DialAug + Reordering	82.7± .2	89.5±.1	19.7±1.0	33.3±1.4	56.2±.6	66.0±.2	87.9±.2	89.8±.1
DialAug + Reordering + CL	82.9±.1	89.6±.1	19.4±.6	33.3±.2	55.8±.4	65.5±.3	88.0±.1	89.9±.1
DialAug + Replacement	82.8±.1	89.5±.1	19.4±.3	31.6±.7	57.5±.7	67.1±.6	89.5±.1	90.9±.1
DialAug + Replacement + CL	82.9±.1	89.6±.1	20.9±.7	33.1±.4	58.0±.7	67.3±.4	89.0±.2	90.6±.1
DialAug + ConMix	83.4±.1	89.9±.0	21.8±1.4	34.9±.4	56.8±.3	66.6±.2	**90.4**±.1	**91.4**±.0
DialAug + ConMix + CL	**83.6**±.1	**90.0**±.0	**23.0**±.8	**36.0**±.7	57.7±.4	67.0±.1	90.1±.3	91.3±.2

Table 2: Results on the Ubuntu V2, Advising DSTC7, Ubuntu DSTC7, and Taskmaster datasets. Results were averaged over three runs, and ± denotes the standard deviation. The numbers in bold denote the best performing model for each dataset.

and can significantly reduce standard deviations of the metrics (1.4 to 0.8 for Recall@1 metric for ConMix augmentation on Advising, and 0.8 to 0.1 from for Subsequence augmentation on Ubuntu).

5.2 Evaluating Robustness to Perturbations

Next we evaluate the data augmentation methods on various perturbations introduced in the dialogue context in the test set. Through this series of experiments we evaluate how robust the model is for different formulations and rewrites of input contexts.

Specifically, we introduce three perturbations that are similar to the augmentation methods used during training:

1. Truncation: Similar to subsequence sampling, we randomly truncate dialogue contexts to remove earlier turns.

2. Word deletion: Delete words with a 30% deletion rate.

3. Word reordering: Reorder words with 30% probability.

We include two additional reformulations that are commonly observed during real-world deployment of models:

4. Typos: We implement the vanilla noise model (Namysl et al., 2020) with noise level 0.1 to capture character-level variations caused by typos. We randomly change 30% of words in the context.

5. Synonym replacement: To capture lexical variations, we randomly replace 30% of words from the context with their synonyms using a pre-defined vocabulary (Jia et al., 2019).

We apply the perturbations independently on the original test sets and evaluate our DialAug model architecture in combination with different training augmentation methods on these harder test sets. As baselines with no augmentations, apart from Bi-encoder, we also include the more powerful Poly-encoder (Humeau et al., 2019) architecture in this evaluation setup.

As can be seen from the results of Table 3, training on augmented data helps significantly against adversarial examples during inference, compared to baseline models trained with no augmentation. It is interesting to note that a more expressive model such as Poly-encoder, with an order of magnitude larger number of parameters, is still susceptible to adversarial perturbations and under-performs the proposed DialAug model that leverages data augmentations. These experiments demonstrate that robustness to noise does not come out-of-the-box for larger models. Instead, strategic data augmentation methods such as ours, that expose a model to diverse training data, can learn to handle these variations effectively.

Comparing among different augmentation methods, it is not surprising to find that a model trained with one augmentation (e.g. subsequence sampling) performs well when exposed to that specific type of perturbations (e.g. truncation) during test. However, they do not generalize well to a different type of noise seen during test (e.g. model trained with deletion based augmentation and tested on reordering). ConMix, on the other hand, is consistently robust to different perturbations across the four adversarial datasets, even though it had not been trained specifically for them. It performs on par or better than the specific data augmentations such as deletion and reordering when exposed to those perturbations during test. For more com-

Dataset: Ubuntu V2										
Augmentation	truncation		deletion		reordering		typo		synonym	
in training	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR
NA (Bi-encoder)	69.0±.2	79.7±.1	69.6±.6	80.2±.0	79.6±.1	87.5±.0	80.8±.1	88.3±.1	79.6±.2	87.4±.1
NA (Poly-encoder)	69.2±.2	79.7±.1	71.1±.3	81.2±.2	80.6±.1	88.0±.0	81.9±.2	88.2±.1	80.7±.3	88.1±.1
Subsequence	**72.1**±.2	**82.1**±.2	68.3±.4	79.0±.0	79.8±.1	87.5±.1	81.1±.2	88.4±.1	79.5±.2	87.4±.1
Deletion	70.0±.1	80.3±.1	**73.1**±.2	**82.8**±.2	80.4±.1	87.9±.1	81.5±.2	88.7±.1	80.4±.1	87.9±.1
Reordering	69.4±.2	79.9±.1	72.2±.1	82.0±.0	80.5±.1	88.0±.0	81.1±.1	88.4±.0	80.5±.1	87.9±.0
Replacement	69.5±.1	79.7±.1	69.6±.5	80.3±.3	79.8±.3	87.5±.1	80.9±.1	88.3±.1	79.7±.1	87.5±.1
ConMix	68.8±.2	79.5±.1	**73.1**±.1	**82.8**±.1	**81.3**±.1	**88.5**±.1	**82.1**±.1	**89.1**±.1	**81.2** ±.0	**88.5**±.0

Dataset: Advising DSTC7										
Augmentation	truncation		deletion		reordering		typo		synonym	
in training	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR
NA (Bi-encoder)	18.4±1.4	29.2±.5	15.5±.1	26.1±.1	14.8±.3	25.7±.4	18.3±.7	30.3±.4	19.6±.8	30.7±.6
NA (Poly-encoder)	17.5±.9	28.5±1.4	**17.9**±1.5	**29.7**±1.3	15.0±.8	26.4±1.0	13.1±.4	25.1±.0	17.0±2.5	28.6±2.3
Subsequence	**19.7**±.1	**31.7**±.1	16.3±.7	27.0±.4	15.4±.8	26.3±.2	18.9±.7	31.1±.5	18.0±.8	29.9±.6
Deletion	18.6±.8	30.4±.1	17.6±.3	29.4±.2	16.6±.0	28.5±.0	19.3±1.3	32.2±.9	18.6±.0	31.8±.4
Reordering	19.0±.0	29.6±.4	15.9±1.8	27.8±1.3	17.1±.1	**30.4**±.5	18.7±1.3	31.9±.6	18.1±1.0	31.0±.3
Replacement	17.7±.8	28.6±.2	14.6±.5	24.8±.9	12.7±.7	23.6±1.1	18.0±.3	29.7±.1	16.3±.1	28.2±.1
ConMix	18.6±.0	29.8±.2	16.2±.6	28.0±.1	**18.3**±.7	30.2±.2	**19.6**±1.4	**32.7**±.5	**20.9**±1.0	**33.2** ±.9

Dataset: Ubuntu DSTC7										
Augmentation	truncation		deletion		reordering		typo		synonym	
in training	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR
NA (Bi-encoder)	42.3±.1	51.7±.3	52.3±.0	61.9±.3	47.9±.6	57.6±.3	48.0±.0	58.2±.4	51.6±.4	61.5±.2
NA (Poly-encoder)	41.2±.8	51.3±.3	49.6±1.1	60.1±.9	47.9±.5	57.5±.2	45.2±.1	56.2±.1	50.6±.4	61.4±.1
Subsequence	**45.7**±.2	**55.8**±.2	47.6±.6	58.3±.2	47.5±.8	58.3±.4	50.8±.3	61.8±.3	52.3±.9	62.6±.4
Deletion	42.4±1.4	52.7±.6	**54.6**±.2	**64.3**±.3	50.3±.3	60.8±.2	53.5±.1	63.5±.1	52.9±.6	63.6±.0
Reordering	41.6±.3	51.2±.6	50.1±.9	59.7±.7	52.8±.3	62.8±.2	51.5±.4	61.7±.3	51.7±.4	62.0±.4
Replacement	43.9±1	53.6±.9	49.7±1	59.9±1	52.1±2.5	62.3±1.7	54.2±.4	**64.4**±.2	**55.0**±.7	**64.7**±.5
ConMix	44.0±.6	52.8±1	50.5±.4	60.8±.5	**54.1**±.5	**63.8**±.3	**54.5**±.4	64.2±.5	54.5±.2	64.1±.2

Dataset: Taskmaster										
Augmentation	truncation		deletion		reordering		typo		synonym	
in training	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR	Rec@1	MRR
NA (Bi-encoder)	76.5±.5	80.8±.4	79.5±.1	84.2±.1	76.5±.2	82.1±.2	87.6±.3	89.6±.2	84.6±.2	87.7±.1
NA (Poly-encoder)	77.1±.0	81.3±.0	79.6±.4	84.2±.2	76.5±.3	82.0±.2	88.0±.1	89.8±.1	84.8±.2	87.8±.1
Subsequence	**85.7**±.2	**88.1**±.1	79.9±.2	84.4±.2	74.5±.8	80.4±.5	87.6±.2	89.5±.1	84.2±.4	87.4±.2
Deletion	76.4±.0	80.6±.0	**86.4**±.2	**88.8**±.1	86.0±.3	88.5±.2	88.5±.2	90.2±.1	86.5±.3	88.9±.2
Reordering	75.9±.3	80.2±.2	83.9±.2	87.2±.1	**89.1**±.3	**90.6**±.2	88.0±.1	89.9±.1	86.3±.1	88.8±.1
Replacement	74.5±.4	79.5±.2	77.0±.5	82.4±.3	71.9±1.0	78.6±.6	86.7±.4	88.4±.3	82.3±.9	86.2±.6
ConMix	81.3±.4	81.3±.3	85.0±.3	87.9±.2	88.2±.3	90.1±.2	**90.1**±.3	**91.3**±.2	**89.3**±.3	**90.8**±.2

Table 3: Robustness during inference for different augmentation strategies. All models using augmentions were trained with contrastive loss. Results were averaged over three runs, and ± denotes the standard deviation.

mon and realistic variations, i.e., synonyms and typos, ConMix significantly outperforms all other methods on three datasets. This indicates a uniformly powerful and robust representation learning method through this novel augmentation strategy.

5.3 Computational Efficiency

ConMix is designed and implemented to generate augmentations through vectorization and therefore has the benefit of being faster to train. Tokens are randomly mixed on-the-fly within a batch to create augmentations in parallel on GPUs, through fast tensor multiplications. For many augmentation methods, such vectorization might be non-trivial and the overall speed becomes limited by the process of creating augmentations outside the training loop on much slower CPUs. For example, when training on the Taskmaster dataset with 8 gpus, the DialAug architecture with ConMix is 1.2x faster than training with the global word replacement augmentation. While conducting full training over 20 epochs this leads to an overall speed up by 1.5 hours for training with the ConMix augmentation.

6 Summary

In this work we proposed DialAug, a modular architecture for conversational response ranking. It combines the traditional cross-entropy loss for ranking with a contrastive counterpart to learn from augmented views of the dialogue context. We presented a novel data augmentation method, ConMix, which generates multiple views of the same context via stochastic mixing of tokens from other contexts in the batch during training. We conducted an extensive set of experiments on four datasets and show that a model trained with ConMix outperforms strong baselines and other augmentation methods. Our proposed model is also proven to be robust against common perturbations encountered during inference. We hope our work encourages further research in such data-centric methods to improve robustness of NLP models for practical applications of conversational modeling.

References

Christopher Beckham, Sina Honari, Vikas Verma, Alex Lamb, Farnoosh Ghadiri, R Devon Hjelm, Yoshua Bengio, and Christopher Pal. 2019. On adversarial mixup resynthesis. *arXiv preprint arXiv:1903.02709*.

B. Byrne, K. Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, A. Cedilnik, and Kyu-Young Kim. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *EMNLP/IJCNLP*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Alexander Richard Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. 2021. Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 704–717.

Hongchao Fang and P. Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *ArXiv*, abs/2005.12766.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, T. Mitamura, and E. Hovy. 2021. A survey of data augmentation approaches for nlp. *ArXiv*, abs/2105.03075.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv*, abs/2104.08821.

John Michael Giorgi, Osvald Nitski, Gary D Bader, and Bo Wang. 2020. Declutr: Deep contrastive learning for unsupervised textual representations. *ArXiv*, abs/2006.03659.

Jia-Chen Gu, Tianda Li, Quan Liu, Xiao-Dan Zhu, Zhenhua Ling, Zhiming Su, and Si Wei. 2020. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

Chulaka Gunasekara, Jonathan K Kummerfeld, Lazaros Polymenakos, and Walter Lasecki. 2019. Dstc7 task 1: Noetic end-to-end response selection. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 60–67.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*.

Janghoon Han, Taesuk Hong, Byoungjae Kim, Youngjoong Ko, and Jungyun Seo. 2021. Fine-grained post-training for improving retrieval-based dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1549–1558.

Matthew Henderson, Ivan Vulic, D. Gerz, I. Casanueva, Paweł Budzianowski, Sam Coope, Georgios P. Spithourakis, Tsung-Hsien Wen, N. Mrksic, and Pei hao Su. 2019. Training neural response selection for task-oriented dialogue systems. In *ACL*.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142.

Mohammad Kachuee, Hao Yuan, Young-Bum Kim, and Sungjin Lee. 2021. Self-supervised contrastive learning for efficient user satisfaction prediction in conversational agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4053–4064.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Ashutosh Kumar, S. Bhattamishra, Manik Bhandari, and P. Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *NAACL-HLT*.

Jonathan K. Kummerfeld, S. R. Gouravajhala, Joseph Peper, V. Athreya, R. Chulaka Gunasekara, Jatin Ganhotra, S. Patel, L. Polymenakos, and Walter S. Lasecki. 2018. Analyzing assumptions in conversation disentanglement research through the lens of a new dataset and model. *ArXiv*, abs/1810.11118.

Shayne Longpre, Yu Wang, and Chris DuBois. 2020. How effective is task-agnostic data augmentation for pretrained transformers? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4401–4411.

Ryan Lowe, Nissan Pow, Iulian Vlad Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.

Y. Lu, Manisha Srivastava, Jared Kramer, Heba Elfardy, Andrea Kahn, Song Wang, and Vikas Bhardwaj. 2019. Goal-oriented end-to-end conversational models with profile features in a real-world setting. In *NAACL*.

Xiaofei Ma, C. D. Santos, and Andrew O. Arnold. 2021. Contrastive fine-tuning improves robustness for neural rankers. In *EMNLP FINDINGS*.

Shikib Mehri, E. Razumovskaia, Tiancheng Zhao, and M. Eskénazi. 2019. Pretraining methods for dialog context representation learning. In *ACL*.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. 2020. Pytorch metric learning.

Marcin Namysl, Sven Behnke, and Joachim Köhler. 2020. Nat: Noise-aware training for robust neural sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1501–1517.

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)*, pages 3982–3992.

Dinghan Shen, Ming Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *ArXiv*, abs/2009.13818.

Connor Shorten and T. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48.

Jason Wei, Chengyu Huang, Soroush Vosoughi, Yu Cheng, and Shiqi Xu. 2021. Few-shot text classification with triplet networks, data augmentation, and curriculum learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5493–5500.

Taesun Whang, Dongyub Lee, Chanhee Lee, Kisu Yang, Dongsuk Oh, and Heuiseok Lim. 2020. An effective domain adaptive post-training method for bert in response selection. In *INTERSPEECH*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *CoRR*, abs/1901.08149.

Chien-Sheng Wu, S. Hoi, R. Socher, and Caiming Xiong. 2020a. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue. In *EMNLP*.

Z. Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020b. Clear: Contrastive learning for sentence representation. *ArXiv*, abs/2012.15466.

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

Ruijian Xu, Chongyang Tao, Daxin Jiang, Xueliang Zhao, Dongyan Zhao, and Rui Yan. 2021. Learning an effective context-response matching model with self-supervised tasks for retrieval-based dialogues. In *AAAI*.