**CoDA: Contrast-enhanced and Diversity-promoting Data Augmentation for Natural Language Understanding**

Yanru Qu\(^1\), Dinghan Shen\(^2\), Yelong Shen\(^2\), Sandra Sajeev\(^2\), Jiawei Han\(^1\), Weizhu Chen\(^2\)

\(^1\)University of Illinois, Urbana-Champaign, \(^2\)Microsoft Dynamics 365 AI

\(^1\)\{yanruqu2,hanj\}@illinois.edu, \(^2\)\{dishen,yeshe,ssajeev,wzchen\}@microsoft.com

**ABSTRACT**

Data augmentation has been demonstrated as an effective strategy for improving model generalization and data efficiency. However, due to the discrete nature of natural language, designing label-preserving transformations for text data tends to be more challenging. In this paper, we propose a novel data augmentation framework dubbed CoDA, which synthesizes diverse and informative augmented examples by integrating multiple transformations organically. Moreover, a contrastive regularization objective is introduced to capture the global relationship among all the data samples. A momentum encoder along with a memory bank is further leveraged to better estimate the contrastive loss. To verify the effectiveness of the proposed framework, we apply CoDA to Transformer-based models on a wide range of natural language understanding tasks. On the GLUE benchmark, CoDA gives rise to an average improvement of 2.2% while applied to the RoBERTa-large model. More importantly, it consistently exhibits stronger results relative to several competitive data augmentation and adversarial training baselines (including the low-resource settings). Extensive experiments show that the proposed contrastive objective can be flexibly combined with various data augmentation approaches to further boost their performance, highlighting the wide applicability of the CoDA framework.

**1 INTRODUCTION**

Data augmentation approaches have successfully improved large-scale neural-network-based models, (Laine & Aila, 2017; Xie et al., 2019; Berthelot et al., 2019; Sohn et al., 2020; He et al., 2020; Khosla et al., 2020; Chen et al., 2020b), however, the majority of existing research is geared towards computer vision tasks. The discrete nature of natural language makes it challenging to design effective label-preserving transformations for text sequences that can help improve model generalization (Hu et al., 2019; Xie et al., 2019). On the other hand, fine-tuning powerful, over-parameterized language models proves to be difficult, especially when there is a limited amount of task-specific data available. It may result in representation collapse (Aghajanyan et al., 2020) or require special finetuning techniques (Sun et al., 2019; Hao et al., 2019). In this work, we aim to take a further step towards finding effective data augmentation strategies through systematic investigation.

In essence, data augmentation can be regarded as constructing neighborhoods around a training instance that preserve the ground-truth label. With such a characterization, adversarial training (Zhu et al., 2020; Jiang et al., 2020; Liu et al., 2020; Cheng et al., 2020) also performs label-preserving transformation in embedding space, and thus is considered as an alternative to data augmentation methods in this work. From this perspective, the goal of developing effective data augmentation strategies can be summarized as answering three fundamental questions:

1) What are some label-preserving transformations, that can be applied to text, to compose useful augmented samples?

\(^1\)Work was done during an internship at Microsoft Dynamics 365 AI.

\(^2\)To name a few, BERT (Devlin et al., 2019): 340M parameters, T5 (Raffel et al., 2019): 11B parameters, GPT-3 (Brown et al., 2020): 175B parameters.
ii) Are these transformations complementary in nature, and can we find some strategies to consolidate them for producing more diverse augmented examples?

iii) How can we incorporate the obtained augmented samples into the training process in an effective and principled manner?

Previous efforts in augmenting text data were mainly focused on answering the first question (Yu et al., 2018; Xie et al., 2019; Kumar et al., 2019; Wei & Zou, 2019; Chen et al., 2020a; Shen et al., 2020). Regarding the second question, different label-preserving transformations have been proposed, but it remains unclear how to integrate them organically. In addition, it has been shown that the diversity of augmented samples plays a vital role in their effectiveness (Xie et al., 2019; Gontijo-Lopes et al., 2020). In the case of image data, several strategies that combine different augmentation methods have been proposed, such as applying multiple transformations sequentially (Cubuk et al., 2018; 2020; Hendrycks et al., 2020), learning data augmentation policies (Cubuk et al., 2018), randomly sampling operations for each data point (Cubuk et al., 2020). However, these methods cannot be naively applied to text data, since the semantic meanings of a sentence are much more sensitive to local perturbations (relative to an image).

As for the third question, consistency training is typically employed to utilize the augmented samples (Laine & Aila, 2017; Hendrycks et al., 2020; Xie et al., 2019; Sohn et al., 2020; Miyato et al., 2018). This method encourages the model predictions to be invariant to certain label-preserving transformations. However, existing approaches only examine a pair of original and augmented samples in isolation, without considering other examples in the entire training set. As a result, the representation of an augmented sample may be closer to those of other training instances, rather than the one it is derived from. Based on this observation, we advocate that, in addition to consistency training, a training objective that can globally capture the intrinsic relationship within the entire set of original and augmented training instances can help leverage augmented examples more effectively.

In this paper, we introduce a novel Contrast-enhanced and Diversity-promoting Data Augmentation (CoDA) framework for natural language understanding. To improve the diversity of augmented samples, we extensively explore different combinations of isolated label-preserving transformations in an unified approach. We find that stacking distinct label-preserving transformations produces particularly informative samples. Specifically, the most diverse and high-quality augmented samples are obtained by stacking an adversarial training module over the back-translation transformation. Besides the consistency-regularized loss for repelling the model to behave consistently within local neighborhoods, we propose a contrastive learning objective to capture the global relationship among the data points in the representation space. We evaluate CoDA on the GLUE benchmark (with RoBERTa (Liu et al., 2019) as the testbed), and CoDA consistently improves the generalization ability of resulting models and gives rise to significant gains relative to the standard fine-tuning procedure. Moreover, our method also outperforms various single data augmentation operations, combination schemes, and other strong baselines. Additional experiments in the low-resource settings and ablation studies further demonstrate the effectiveness of this framework.

2 Method

In this section, we focus our discussion on the natural language understanding (NLU) tasks, and particularly, under a text classification scenario. However, the proposed data augmentation framework can be readily extended to other NLP tasks as well.

2.1 Background: Data Augmentation and Adversarial Training

Data Augmentation Let \( \mathcal{D} = \{x_i, y_i\}_{i=1}^N \) denote the training dataset, where the input example \( x_i \) is a sequence of tokens, and \( y_i \) is the corresponding label. To improve model’s robustness and generalization ability, several data augmentation techniques (e.g., back-translation (Sennrich et al., 2016) [Edunov et al., 2018] [Xie et al., 2019], mixup (Guo et al., 2019), c-BERT (Wu et al., 2019)) have been proposed. Concretely, label-preserving transformations are performed (on the original training sequences) to synthesize a collection of augmented samples, denoted by \( \mathcal{D}' = \{x'_i, y'_i\}_{i=1}^N \). Thus, a model can learn from both the training set \( \mathcal{D} \) and the augmented set \( \mathcal{D}' \), with \( p_0(\cdot) \) the predicted output distribution of the model parameterized by \( \theta \):

\[
\theta^* = \arg\min_{\theta} \sum_{(x_i, y_i) \in \mathcal{D}} \mathcal{L}(p_0(x_i), y_i) + \sum_{(x'_i, y'_i) \in \mathcal{D}'} \mathcal{L}(p_0(x'_i), y'_i)
\]  

(1)
Several recent research efforts were focused on encouraging model predictions to be invariant to stochastic or domain-specific data transformations \cite{Xie2019, Laine2017, Tarvainen2017, Sohn2020, Miyato2018, Jiang2020, Hendrycks2020, Cubuk2018, 2020}.

Take back-translation as example: \( x'_i = \text{BackTrans}(x_i) \), then \( x'_i \) is a paraphrase of \( x_i \). The model can be regularized to have consistent predictions for \( (x_i, x'_i) \), by minimizing the distribution discrepancy \( R_{CS}(p_0(x_i), p_0(x'_i)) \), which typically adopts KL divergence (see Fig. 1a).

**Adversarial Training** In another line, adversarial training methods are applied to text data \cite{Zhu2020, Jiang2020, Cheng2020, Aghajanyan2020} for improving model’s robustness. Compared with data augmentation techniques, adversarial training requires no domain knowledge to generate additional training examples. Instead, it relies on the model itself to produce adversarial examples which the model are most likely to make incorrect predictions. Similar to data augmentation, adversarial training also typically utilizes the cross-entropy and consistency-based objectives for training. As the two most popular adversarial-training-based algorithms, the adversarial loss \cite{Goodfellow2015} (Eqn. 2) and virtual adversarial loss \cite{Miyato2018} (Eqn. 3) can be expressed as follows (see Fig. 1b):

\[
R_{AT}(x_i, \bar{x}_i, y_i) = \mathcal{L}(p_0(\bar{x}_i), y_i), \ s.t., \|\bar{x}_i - x_i\| \leq \epsilon, \quad (2)
\]

\[
R_{VAT}(x_i, \bar{x}_i) = R_{CS}(p_0(\bar{x}_i), p_0(x_i)), \ s.t., \|\bar{x}_i - x_i\| \leq \epsilon. \quad (3)
\]

Generally, there is no closed-form to obtain the exact adversarial example \( \bar{x}_i \) in either Eqn. 2 or 3. However, it usually can be approximated by a low-order approximation of the objective function with respect to \( x_i \). For example, the adversarial example in Eqn. 2 can be approximated by:

\[
\bar{x}_i \approx x_i + \epsilon \frac{g}{\|g\|_2}, \text{ where } g = \nabla_{x_i} \mathcal{L}(p_0(x_i), y_i). \quad (4)
\]

2.2 **Diversity-promoting Consistency Training**

As discussed in the previous section, data augmentation and adversarial training share the same intuition of producing neighbors around the original training instances. Moreover, both approaches share very similar training objectives. Therefore, it is natural to ask the following question: are different data augmentation methods and adversarial training equal in nature? Otherwise, are they complementary to each other, and thus can be consolidated together to further improve the model’s generalization ability? Notably, it has been shown, in the CV domain, that combining different data augmentation operations could lead to more diverse augmented examples \cite{Cubuk2018, Hendrycks2020}. However, this is especially challenging for natural language, given that the semantics of a sentence can be entirely altered by slight perturbations.

To answer the above question, we propose several distinct strategies to combine different data transformations, with the hope to produce more diverse and informative augmented examples. Specifically, we consider 5 different types of label-preserving transformations: *back-translation* \cite{Sennrich2016, Edunov2018, Xie2019}, *c-BERT* word replacement \cite{Wu2019}, *mixup* \cite{Guo2019, Chen2020}, *cutoff* \cite{Shen2020}, and *adversarial training* \cite{Zhu2020, Jiang2020}. The 3 combination strategies are schematically illustrated.
Consistency loss only provides local regularization, i.e., $x_i$ and $x'_i$ should have close predictions. However, the relative positions between $x'_i$ and other training instances $x_j$ (j ≠ i) have not been examined. In this regard, we propose to leverage a contrastive learning objective to better utilize the augmented examples. Specifically, we assume that the model should encourage an augmented sample $x'_i$ to be closer, in the representation space, to its original $x_i$, relative to other data points $x_j$ (j ≠ i) in the training set. This is a reasonable assumption since intuitively, the model should be robust enough to successfully determine from which original data an augmented sample is produced.

The contrastive learning module is illustrated in Fig. 3. As demonstrated by prior efforts on contrastive learning, adopting a large batch size is especially vital for its effectiveness ([Chen et al., 2020b; Khosla et al., 2020]). Therefore, we introduce a memory bank that stores the history embeddings, thus enabling much larger number of negative samples. Moreover, to avoid the encoder from changing too rapidly (which may result in inconsistency embeddings), a momentum encoder module is incorporated into our algorithm. Concretely, let $f_{θ}(·)$ and $f_{θ}(·)$ denote the transformation...
parameterized by the query encoder and key encoder, respectively. Note that $\theta$ and $\bar{\theta}$ represent their parameters. The momentum model parameters $\bar{\theta}$ are not learned by gradients. Instead, they are updated through the momentum rule: $\bar{\theta} \leftarrow \gamma \bar{\theta} + (1 - \gamma)\theta$ at each training step. We omit the details here and refer the interested readers to the work by (He et al., 2020) for further explanation. Given a sample $x_i$ and its augmented example $x'_i$, the query and key can be obtained as follows:

$$
q_i = f_\theta(x_i), \quad q'_i = f_\bar{\theta}(x'_i), \quad k_i = f_\bar{\theta}(x_i).
$$

Thus, the contrastive training objective can be written as:

$$
\mathcal{R}_{\text{contrast}}(x_i, x'_i, \mathcal{M}) = \mathcal{R}_{\text{CT}}(q_i, k_i, \mathcal{M}) + \mathcal{R}_{\text{CT}}(q'_i, k_i, \mathcal{M}),
$$

$$
\mathcal{R}_{\text{CT}}(q_i, k_i, \mathcal{M}) = -\log \frac{\exp(\text{sim}(q_i, k_i)/\tau)}{\sum_{k_j \in \mathcal{M} \cup \{k_i\}} \exp(\text{sim}(q_i, k_j)/\tau)},
$$

where $\tau$ is the temperature, and $\mathcal{M}$ is the memory bank in which the history keys are stored. Cosine similarity is chosen for $\text{sim}(\cdot)$. Note that $\mathcal{R}_{\text{CT}}(q'_i, k_i, \mathcal{M})$ is similarly defined as $\mathcal{R}_{\text{CT}}(q_i, k_i, \mathcal{M})$ (with $q_i$ replaced by $q'_i$ in Eqn. 10). In Eqn. 9 the first term corresponds to the contrastive loss calculated on the original examples (self-contrastive loss), while the second term is computed on the augmented sample (augment-contrastive loss). Under such a framework, the pair of original and augmented samples are encouraged to stay closer in the learned embedding space, relative to all other training instances. As a result, the model is regularized globally through considering the embeddings of all the training examples available.

By integrating both the consistency training objective and the contrastive regularization, the overall training objective for the CoDA framework can be expressed as:

$$
\theta^* = \arg\min_\theta \sum_{(x_i, y_i) \in \mathcal{D}} \mathcal{L}_{\text{consistency}}(x_i, x'_i, y_i) + \lambda \mathcal{R}_{\text{contrast}}(x_i, x'_i, \mathcal{M}).
$$

where $\lambda$ is a hyperparameter to be chosen. It is worth noting that the final objective has taken both the local (consistency loss) and global (contrastive loss) information introduced by the augmented examples into consideration.

3 Experiments

To verify the effectiveness of CoDA, We evaluate it on the widely-adopted GLUE benchmark (Wang et al., 2018), which consists of multiple natural language understanding (NLU) tasks. The details of these datasets can be found in Appendix [B]. RoBERTa (Liu et al., 2019) is employed as the testbed for our experiments. However, the proposed approach can be flexibly integrated with other models as well. We provide more implementation details in Appendix [C]. Our code will be released to encourage future research.

In this section, we first present our exploration of several different strategies to consolidate various data transformations (Sec 3.1). Next, we conduct extensive experiments to carefully select the contrastive objective for NLU problems in Sec 3.2. Based upon these settings, we further evaluate CoDA on the GLUE benchmark and compare it with a set of competitive baselines in Sec 3.3. Additional experiments in the low-resource settings and qualitative analysis (Sec 3.4) are further conducted to gain a deep understanding of the proposed framework.

3.1 Combining Label-preserving Transformations

We start by implementing and comparing several data augmentation baselines. As described in the previous section, we explore 5 different approaches: back-translation, c-BERT word replacement, Mixup, Cutoff and adversarial training. More details can be found in Appendix [A]. The standard cross-entropy loss, along with the consistency regularization term (Eq. 4) is utilized for all methods to ensure a fair comparison. We employ the MNLI dataset and RoBERTa-base model for the comparison experiments with the results shown in Table [1].

All these methods have achieved improvements over the RoBERTa-base model, demonstrating the effectiveness of leveraging label-preserving transformations for NLU. Moreover, back-translation, cutoff and adversarial training exhibit stronger empirical results relative to mixup and c-BERT.

To improve the diversity of augmented examples, we explore several strategies to combine multiple transformations: i) random combination, ii) mixup interpolation, and iii) sequential stacking, as
shown in Fig. 2. In Table 1, the score of naive random combination lies between single transformations. This may be attributed to the fact that different label-preserving transformations regularize the model in distinct ways, and thus the model may not be able to leverage different regularization terms simultaneously.

Besides, among all the other combination strategies, we observe that gains can be obtained by integrating back-translation and adversarial training together. Concretely, mixing back-translation and adversarial training samples (in the input embedding space) slightly improve the accuracy from 88.5 to 88.6. More importantly, the result is further improved to 88.8 with these two transformations stacking together (see Sec 2.2). With significance test, we find stack (back, adv) performs consistently better than other combinations (t-test of 10 runs, p-values < 0.02). This observation indicates that the stacking operation, especially in the case of back-translation and adversarial training, can produce more diverse augment examples.

Intuitively, the augmented sample, with two sequential transformations, deviates more from the corresponding training data, and thus tends to be more effective at improving the model’s generalization ability. To verify this hypothesis, we further calculate the MMD (Gretton et al., 2012) between augmented samples and the original training instances. It can be observed that the stacking operation, especially in the case of back-translation and adversarial training, can produce more diverse augment examples.

Table 1: Comparison of different transformations on the MNLI-m development set. Abbr: original training instances (ori), back-translation (back), cutoff (cut), mixup (mix), adversarial (adv).

<table>
<thead>
<tr>
<th>Method</th>
<th>MNLI-m (Acc)</th>
<th>MMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoBERTa-base</td>
<td>87.6</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Single Transformation</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+ back-translation</td>
<td>88.5</td>
<td>0.63</td>
</tr>
<tr>
<td>+ c-BERT</td>
<td>88.0</td>
<td>0.01</td>
</tr>
<tr>
<td>+ cutoff</td>
<td>88.4</td>
<td>0.02</td>
</tr>
<tr>
<td>+ mixup (ori, ori)</td>
<td>88.2</td>
<td>0.06</td>
</tr>
<tr>
<td>+ adversarial</td>
<td>88.5</td>
<td>0.65</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multiple Transformations</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+ random (back, cut, adv)</td>
<td>88.4</td>
<td>-</td>
</tr>
<tr>
<td>+ mix (ori, back)</td>
<td>88.4</td>
<td>0.11</td>
</tr>
<tr>
<td>+ mix (back, adv)</td>
<td>88.6</td>
<td>0.81</td>
</tr>
<tr>
<td>+ stack (back, cut)</td>
<td>88.5</td>
<td>0.62</td>
</tr>
<tr>
<td>+ stack (back, adv)</td>
<td><strong>88.8</strong></td>
<td>1.14</td>
</tr>
<tr>
<td>+ stack (back, cut, adv)</td>
<td>88.5</td>
<td>1.14</td>
</tr>
<tr>
<td>+ stack (back, adv, cut)</td>
<td>88.4</td>
<td>1.14</td>
</tr>
</tbody>
</table>

3.2 CONTRASTIVE REGULARIZATION DESIGN

In this section, we aim to incorporate the global information among the entire set of original and augmented samples via a contrastive regularization. First, we explore a few hyperparameters for the proposed contrastive objective. Since both the memory bank and the momentum encoder are vital components, we study the impacts of different hyperparameter values on both the temperature and the momentum. As shown in Fig. 4, a temperature of 1.0 combined with the momentum of 0.99 can achieve the best empirical result. We then examine the size effect of the memory bank, and observe a larger memory bank size leads to a better capture of the global information and results in higher performance boost.

After carefully choosing the best setting based on the above experiments, we apply the contrastive learning objective to several GLUE datasets. We also implement several prior works on contrastive learning to compare, including the MoCo loss (He et al. 2020) and the supervised contrastive (SupCon) loss (Khosla et al. 2020), all implemented with memory banks. Note that we remove the consistency regularization for this experiment to better examine the effect of the contrastive regularization term (i.e., $\alpha = \beta = 0$, $\lambda \neq 0$). As presented in Table 2, our contrastive objective consistently exhibits the largest performance improvement. This observation demonstrates for NLU, our data transformation module can be effectively equipped with the contrastive regularization.

---

2 In practice, we can use Machine Translation (MT) models trained on large parallel corpus (e.g., English-French, English-German) to back-translate the input sentence. Since back-translation requires decoding, it can be performed offline. If the input contains multiple sentences, we split it into sentences, perform back-translation, and ensemble those paraphrases back.

3 We set the default memory bank size as 65536. As to smaller datasets, we choose the size no larger than the number of training data (e.g., MRPC has 3.7k samples, and we set the memory size as 2048).
Figure 4: Hyperparameter exploration for the contrastive loss, evaluated on the MNLI-m development set. \textit{Note: All models use the RoBERTa-base model as the encoder.}

Table 2: Comparison among different contrastive objectives on the GLUE development set.

3.3 GLUE Benchmark Evaluation

With both components within the CoDA algorithm being specifically tailored to the natural language understanding applications, we apply it to the RoBERTa-large model \cite{Liu2019RoBERTa}. Comparisons are made with several competitive data-augmentation-based and adversarial-training-based approaches on the GLUE benchmark. Specifically, we consider back-translation, cutoff \cite{Shen2020Cutoff}, FreeLB \cite{Zhu2020FreeLB}, SMART \cite{Jiang2020SMART}, and R3F \cite{Aghajanyan2020R3F} as the baselines, where the last three all belong to adversarial training. The results are presented in Table 3. It is worth noting that back-translation is based on our implementation, where both the cross-entropy and consistency regularization terms are utilized.

Table 3: Main results of single models on the GLUE development set. \textit{Note: The best result on each task is in \textbf{bold} and “-” denotes the missing results. The average score is calculated based on the same setting as RoBERTa.}

We find that CoDA brings significant gains to the RoBERTa-large model, with the averaged score on the GLUE dev set improved from 88.9 to 91.1. More importantly, CoDA consistently outperforms these strong baselines (indicated by a higher averaged score), demonstrating that our algorithm can produce informative and high-quality augmented samples and leverage them effectively as well. Concretely, on datasets with relatively larger numbers of training instances (> 100K), \textit{i.e.}, MNLI, QQP and QNLI, different approaches show similar gains over the RoBERTa-large model. However, on smaller tasks (SST-2, MRPC, CoLA, RTE, and STS-B), CoDA beats other data augmentation or adversarial-based methods by a wide margin. We attribute this observation to the fact that the synthetically produced examples are more helpful when the tasks-specific data is limited. Thus, when smaller datasets are employed for fine-tuning large-scale language models, the superiority of the proposed approach is manifested to a larger extent.

3.4 Additional Experiments and Analysis

Low-resource Setting To verify the advantages of CoDA when a smaller number of task-specific data is available, we further conduct a low-resource experiment with the MNLI
and QNLI datasets. Concretely, different proportions of training data are sampled and utilized for training. We apply CoDA to RoBERTa-base and compare it with back-translation and adversarial training across various training set sizes. The corresponding results are presented in Fig. 5. We observe that back-translation and adversarial training exhibit similar performance across different proportions. More importantly, CoDA demonstrates stronger results consistently, further highlighting its effectiveness with limited training data.

The Effectiveness of Contrastive Objective To investigate the general applicability of the proposed contrastive regularization objective, we further apply it to different data augmentation methods. The RoBERTa-base model and QNLI dataset are leveraged for this set of experiments, and the results are shown in Fig. 6. We observe that the contrastive learning objective boosts the empirical performance of the resulting algorithm regardless of the data augmentation approaches it is applied to. This further validates our assumption that considering the global information among the embeddings of all examples is beneficial for leveraging augmented samples more effectively.

4 RELATED WORK

Data Augmentation in NLP Different data augmentation approaches have been proposed for text data, such as back-translation (Sennrich et al., 2016; Edunov et al., 2018; Xie et al., 2019), c-BERT word replacement (Wu et al., 2019), mixup (Guo et al., 2019; Chen et al., 2020a), Cutoff (Shen et al., 2020). Broadly speaking, adversarial training (Zhu et al., 2020; Jiang et al., 2020) also synthesizes additional examples via perturbations at the word embedding layer. Although effective, how these data augmentation transformations may be combined together to obtain further improvement has been rarely explored. This could be attributed to the fact that a sentence’s semantic meanings are quite sensitive to small perturbations. Consistency-regularized loss (Bachman et al., 2014; Rasmus et al., 2015; Laine & Aila, 2017; Tarvainen & Valpola, 2017) is typically employed as the training objective, which ignores the global information within the entire dataset.

Contrastive Learning Contrastive methods learn representations by contrasting positive and negative examples, which has demonstrated impressive empirical success in computer vision tasks (Hénaff et al., 2019; He et al., 2020). Under an unsupervised setting, Contrastive learning approaches learn representation by maximizing mutual information between local-global hidden representations (Hjelm et al., 2019; Oord et al., 2018; Hénaff et al., 2019). It can be also leveraged to learn invariant representations by encouraging consensus between augmented samples from the same input (Bachman et al., 2019; Tian et al., 2019; He et al., 2020; Wu et al., 2018). proposes to utilize a memory bank to enable a much larger number of negative samples, which is shown to benefit the transferability of learned representations as well (Khosla et al., 2020). Recently, contrastive learning was also employed to improve language model pre-training (He et al., 2020).

5 CONCLUSION

In this paper, we proposed CoDA, a Contrast-enhanced and Diversity promoting data Augmentation framework. Through extensive experiments, we found that stacking adversarial training over a back-translation module can give rise to more diverse and informative augmented samples. Besides,
we introduced a specially-designed contrastive loss to incorporate these examples for training in a principled manner. Experiments on the GLUE benchmark showed that CoDA consistently improves over several competitive data augmentation and adversarial training baselines. Moreover, it is observed that the proposed contrastive objective can be leveraged to improve other data augmentation approaches as well, highlighting the wide applicability of the CoDA framework.

REFERENCES


A DATA AUGMENTATION DETAILS

We select the following representative data augmentation operations as basic building blocks of our data augmentation module. We denote $x_i = [x_{i,1}, \ldots, x_{i,l}]$ as the input text sequence, and $e_i = [e_{i,1}, \ldots, e_{i,l}]$ as corresponding embedding vectors.

- Back translation is widely applied in machine translation (MT) [Sennrich et al., 2016; Hoang et al., 2018; Edunov et al., 2018], and is introduced to text classification recently [Xie et al., 2019]. Back-Trans uses 2 MT models to translate the input example to another pivot language, and then translate it back, $x_i \rightarrow \text{Pivot Language} \rightarrow x'_i$.

- C-BERT Word Replacement [Wu et al., 2019] is a representative of the word replacement augmentation family. C-BERT pretrains a conditional BERT model to learn contextualized representation $P(x_j | [x_{i,1} \ldots x_{i,j-1} | \text{MASK}] x_{i,j+1} \ldots x_{i,l}, y_i)$ conditioning on classes. This method then randomly substitutes words of $x$ to obtain $x'_i ([x_{i,1} \ldots x'_{i,j} \ldots x_{i,l}])$.

- Cutoff [DeVries & Taylor, 2017] randomly drops units in a continuous span on the input, while Shen et al. (2020) adapts this method to text embeddings. For input embeddings $e_i$, this method randomly set a continuous span of elements to 0s, $e'_i = [e_{i,1} \ldots e_{i,j-1}, 0 \ldots 0, e_{i,j+w} \ldots e_{i,l}]$, where the window size $w \propto l$, and the start position $j \in [1, l-w]$ is randomly selected. For transformer encoders that involve position embeddings, we also set input mask as 0s at corresponding positions.

- Mixup [Zhang et al., 2017] interpolates two image as well as their labels. Guo et al. (2019) borrows this method to text. For 2 input embeddings $(e_i, e_j)$, mixup interpolates the embedding vectors $v'_i = ae_i + (1-a)e_j$ where $a$ is sampled from a Beta distribution. Also, the labels are interpolated for the augmented sample $y'_i = ay_i + (1-a)y_j$.

- Adversarial training generates adversarial examples for input embeddings, simply, $e'_i = \arg \max_{\|e_i - e'_i\| \leq L} L(f(e'_i), y_i)$. We mainly follow the implementation of Zhu et al. (2020). Besides, when computing the adversarial example $e'_i$, the dropout variables are recorded and reused later when encoding $e'_i$.

Maximum mean discrepancy (MMD) [Gretton et al., 2012] is a widely used discrepancy measure for 2 distributions. We adopt the multi-kernel MMD implementation based on Shen et al. (2018) to quantify the distance of data distributions before and after DA transformations.

B DATASET DETAILS

The datasets and statistics are summarized in Table 4.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Task</th>
<th>Sentence Pair</th>
<th>#Train</th>
<th>#Dev</th>
<th>#Test</th>
<th>#Class</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNLI</td>
<td>NLI</td>
<td>✓</td>
<td>393k</td>
<td>20k</td>
<td>20k</td>
<td>3</td>
<td>Accuracy</td>
</tr>
<tr>
<td>QQP</td>
<td>Paraphrase</td>
<td>✓</td>
<td>364k</td>
<td>40k</td>
<td>391k</td>
<td>2</td>
<td>Accuracy/F1</td>
</tr>
<tr>
<td>QNLI</td>
<td>QA/NLI</td>
<td>✓</td>
<td>108k</td>
<td>5.7k</td>
<td>5.7k</td>
<td>2</td>
<td>Accuracy</td>
</tr>
<tr>
<td>SST</td>
<td>Sentiment</td>
<td>×</td>
<td>6.7k</td>
<td>872</td>
<td>1.8k</td>
<td>2</td>
<td>Accuracy</td>
</tr>
<tr>
<td>MRPC</td>
<td>Paraphrase</td>
<td>✓</td>
<td>3.7k</td>
<td>408</td>
<td>1.7k</td>
<td>2</td>
<td>Accuracy/F1</td>
</tr>
<tr>
<td>CoLA</td>
<td>Acceptability</td>
<td>×</td>
<td>8.5k</td>
<td>1k</td>
<td>1k</td>
<td>2</td>
<td>Matthews corr</td>
</tr>
<tr>
<td>RTE</td>
<td>NLI</td>
<td>✓</td>
<td>2.5k</td>
<td>276</td>
<td>3k</td>
<td>2</td>
<td>Accuracy</td>
</tr>
<tr>
<td>STS-B</td>
<td>Similarity</td>
<td>✓</td>
<td>7k</td>
<td>1.5k</td>
<td>1.4k</td>
<td>-</td>
<td>Pearson/Spearman corr</td>
</tr>
</tbody>
</table>

Table 4: GLUE benchmark summary.

---

4EDA [Wei & Zou, 2019] uses synonym replacement, another word replacement technique. We choose C-BERT for this family to take the advantages of contextual representation.

5https://github.com/RockySJ/WDGRL
C IMPLEMENTATION DETAILS

Our implementation is based on RoBERTa (Liu et al., 2019). We use ADAM (Kingma & Ba, 2014) as our optimizer. We follow the hyper-parameter study of RoBERTa and set as default the following parameters: batch size (32), learning rate (1e-5), epochs (5), warmup ratio (0.06), weight decay (0.1) and we keep other parameters unchanged with RoBERTa. For Back-Trans, we use the en-de single models trained on WMT19 and released in FairSeq. More specifically, we use beam search (beam size = 5) and keep only the top-1 hypothesis. We slightly tune Adversarial parameters on MNLI based on FreeLB and fix them on other datasets, since adversarial training is not our focus. For contrastive regularization, we implement based on MoCo. In GLUE evaluation, we mainly tune the weights of 3 regularization terms, $\alpha \in [0, 1]$, $\beta \in [0, 3]$, $\lambda \in [0, 0.03]$ (Eq. 6, 11). Besides, for smaller tasks (MRPC, CoLA, RTE, STS-B), we use the best performed MNLI model to initialize their parameters.\footnote{RoBERTa: https://github.com/huggingface/transformers. FairSeq: https://github.com/pytorch/fairseq. FreeLB: https://github.com/zhuchen03/FreeLB. MoCo: https://github.com/facebookresearch/moco. We will release our model and code for further study.}