

# A Diversity-Enhanced and Constraints-Relaxed Augmentation for Low-Resource Classification

Guang Liu, Hailong Huang, Yuzhao Mao, Weiguo Gao, Xuan Li, and Jianping Shen

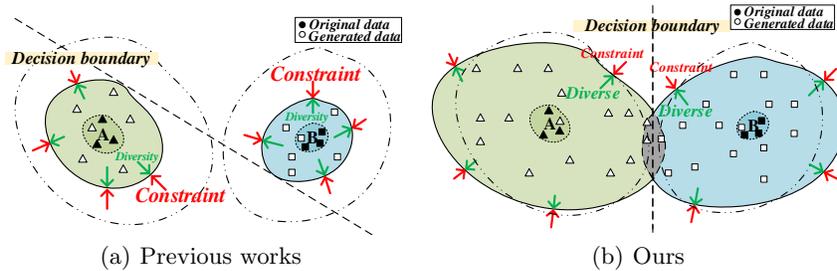
PingAn Life Insurance of China

**Abstract.** Data augmentation (DA) aims to generate constrained and diversified data to improve classifiers in Low-Resource Classification (LRC). Previous studies mostly use a fine-tuned Language Model (LM) to strengthen the constraints but ignore the fact that the potential of diversity could improve the effectiveness of generated data. In LRC, strong constraints but weak diversity in DA result in the poor generalization ability of classifiers. To address this dilemma, we propose a **Diversity-Enhanced and Constraints-Relaxed Augmentation (DECRA)**. Our DECRA has two essential components on top of a transformer-based backbone model. 1) A  **$k$ - $\beta$  augmentation**, an essential component of DECRA, is proposed to enhance the diversity in generating constrained data. It expands the changing scope and improves the degree of complexity of the generated data. 2) A masked language model loss, instead of fine-tuning, is used as a **regularization**. It relaxes constraints so that the classifier can be trained with more scattered generated data. The combination of these two components generates data that can reach or approach category boundaries and hence help the classifier generalize better. We evaluate our DECRA on three public benchmark datasets under low-resource settings. Extensive experiments demonstrate that our DECRA outperforms state-of-the-art approaches by 3.8% in the overall score.

**Keywords:** text mining · data augmentation · regularization · low-resource classification.

## 1 Introduction

Data Augmentation (DA) approaches [7, 3, 13, 15, 35] are often used to alleviate the thirst for labeled data in Low-Resource Classification (LRC [29]). Classification is essential in building intelligent systems, such as empathetic dialogue systems [39, 5] and medical diagnosis systems [1, 20]. In most cases, external resources, such as the similar task data [35] and unlabeled data [33], are hard to obtain or even unavailable. In low-resource settings, the greatest challenge is the expensive cost as well as the shortage of experienced experts who serve to collect and label large-scale data. Without sufficient labeled data, the deep networks tend to generalize poorly, leading to unsatisfactory performance.



**Fig. 1.** The demonstration of how augmentation works in LRC. *Fig 1(a)* is the augmentation with strong constraints and weak diversity. The generated data is near to the original ones. So, the classifier learns a decision boundary that generalizes poorly. *Fig 1(b)* is augmentation with enhanced diversity and relaxed constraints. Ideally, the generated data will be close to the boundary of the categories. Therefore, the classifier learns a decision boundary that has a better generalization ability.

Data Augmentation (DA) in text data aims to generate constrained and diversified data to improve classifier performance [38, 11, 14, 15]. Ideally, we assume data generated by DA is able to present the data distribution in every category. Thus, the generated data is supposed to extend the range of labeled data which would help the classifier make better decisions [36]. As shown in Fig. 1(a), constraints and diversity are two main concepts in DA. The constraints mainly pull the generated data towards the original one. The diversity in generating comes from partially changed labeled data. It pushes the generated data away from the original one. Previous researches focus on strengthening the constraints [18, 37, 15], especially contextually. The fine-tuned [2, 12] Language Models (LM) are often used to generate contextual constraints, e.g., Bidirectional Encoder Representations from Transformers (BERT[8]). In the meantime, additional constraints from the labels are introduced through fine-tuning, such as the Conditional BERT (CBERT [18]), which is fine-tuned with the additional constraints on labels. CBERT overfits in the low data conditions and consequently lacks diversity in generating. To improve generalization ability, the Learning Data Manipulate for Augmentation and Weighting (LDMAW [15]) unifies the learning targets of both the augmentation and the classification through a reinforcement learning framework. Noticeably, LDMAW uses a BERT (fine-tuned LM) to generate data for another BERT (classifier).

As depicted in Fig. 1(a), previous studies often suffer from poor generalization ability in low-resource conditions [33] due to strong constraints but weak diversity in augmentation. As the Language Model (LM) tends to overfit on limited data in low-resource conditions, strong constraints are formed after fine-tuning [37, 15]. As a result, the generated data is pulled towards the original data. At the same time, the weakness of diversity in augmentation is often ignored. Current DA approaches mostly use the method that is identical to the masked Language Model learning in BERT [8]. In this method, diversity is in-

fluenced by the changing scope and degree of complexity in the generated data. The changing scope is proportional to the times of the DA applied. In each time of augmentation, one set of maskers is generated [15, 18]. And the masked positions are the ones to be augmented. This process results in a fixed and narrow changing scope. On the other hand, the degree of complexity is related to the amount of information used in the augmenting data in masked positions. For each masked position, routinely, one sampled tokens are applied [37, 18]. Therefore, it results in the low complexity of the generated data. Consequently, strong constraints but weak diversity causes the poor generalization ability in LRC.

To address the described problem, we propose a **Diversity-Enhanced and Constraints-Relaxed Augmentation (DECRA)**, as displayed in Fig. 1(b). DECRA allows the generated data to be more scattered within the extended boundary. Our DECRA is based on the modified LDMAW [15], which is the state-of-the-art model in LRC. The backbone model, a simplified LDMAW, shares parameters in BERT to reduce overfitting for better generalization ability. The backbone model consists of a transformer-based encoder (TBE), a language model layer (LML) and a classification layer (CL). DECRA has two essential components based on the backbone model:  $k$ - $\beta$  augmentation and regularization (masked LM loss). 1)  $k$ - $\beta$  augmentation, an essential component in DECRA, will enhance the diversity in generating. It expands the changing scope by applying augmentation  $\beta$  times and enhances the degree of complexity by using top- $k$  tokens to augment the masked position. 2) The regularization, masked LM loss on original data, generates more relaxed constraints compared to fine-tuning. DECRA will be trained by the combination of masked LM loss and the classification loss. Our model can learn the constraints dynamically and progressively during the training process. It will process more scattered generated data, which will reach or approach the boundary of categories, to achieve better generalization ability. Therefore, enhanced diversity as well as relaxed constraints help to generate data more scattered within the extended boundary. Trained with the labeled and generated data, the classifier will make better decisions and consequently achieve better generalization ability in LRC.

We evaluate DECRA on three text classification benchmark datasets under low-resource settings. Extensive experiments show that our model achieves superior performance than advanced baselines, such as LDMAW and CBERT.

The major contributions of this paper are summarized below:

- 1) We first propose a **Diversity-Enhanced and Constraints-Relaxed Augmentation (DECRA)** for Low-Resource Classification (LRC). Experimental results show that our DECRA outperforms the state-of-the-art approach by 3.8% in the overall score.
- 2) We propose a  $k$ - $\beta$  augmentation to enhance the diversity in constrained generating. It can improve diversity by expanding the changing scope and enhancing the degree of complexity.
- 3) We propose to use the masked Language Model (LM) loss on original data as a regularization instead of fine-tuning. It helps to relax the constraints, and eventually improve the generalization ability of classifiers.

## 2 Related Work

### 2.1 Language model

Recently, many works have shown that pre-trained Language models (LM) on large corpora can learn common language representations, which is beneficial for downstream Natural Language Processing (NLP) tasks and can avoid training new models from scratch [24, 8]. With the development of computing power, the emergence of deep models (i.e. Transformer [34]) and the continuous improvement of training skills, the architecture of LM has evolved from shallow to deep. The first generation of LM is designed to learn contextual-free word embedding. They are usually shallow for computational efficiency [24, 26]. Although these pre-trained embeddings can capture the semantics of words, they have no context and cannot capture high-level concepts in the context. The second generation of LM focuses on learning contextual word embeddings, such as GPT-2 [27] and BERT [8].

### 2.2 Text data augmentation

Data Augmentation (DA) in text data, different from the image data [7, 41], is difficult due to the to preserve grammar and semantics. The text augmentation can be divided into the rule-based approaches and the Language Model (LM)-based approaches.

The rule-based approaches mainly augment labeled data with the prior rules [28, 36, 10]. Some works inject small perturbation into the representation of labeled data [25]. That increases the model’s generalization ability. Some works are inspired by the smoothing hypothesis [32, 31]. They propose to use a weight to mix two labeled data into one generated data [40, 10]. Training with both data, it achieves better generalization ability [6, 4]. Some works use a pre-trained translation model to augment the labeled data by translating it into another language and translate it back to its original language [29].

The LM-based approaches use the Language Model (LM) to generate diverse data with constraints. The essential operation is to randomly replace some tokens in the labeled data with contextual and label constraints. Easy Data Augmentation (EDA) [36] generates data without label constraints which often results in the label-drift. That will introduce noise into the generated data. Staged fine-tuned LM on labeled data is suitable for the operation [23]. The Contextual Augmentation (CA) [18] uses an LSTM based LM to improve the contextual constraints. The LSTM-based LM with context-free word embeddings can not handle the contextual constraints well. Therefore, the contextual aware embeddings are introduced into DA [37, 2].

## 3 Problem formulation

For a text classification dataset  $\{(\mathbf{x}_i, y_i)\}, i \in [1, N]$ , where  $\mathbf{x}_i \in \mathbb{R}^{T \times V}$  and  $y_i \in \mathbb{R}^C$ ,  $N$  is the training data size,  $T$  is the length of data and  $V$  is the vocabulary

size,  $C$  is the number of classes. In Low-Resource Classification (LRC), the  $N$  is very small, such as lower than 40 samples per class. That embodies the needs of Language Model (LM)  $g_{\theta_a}$  to generate diverse data to improve the generalization ability of the classifier  $f_{\theta_c}$ , where  $\theta_a$  and  $\theta_c$  represent the parameters respectively. The generated data should contain constraints as well as maintain diversity to ensure generalization ability. The formulation of an operation in augmentation is

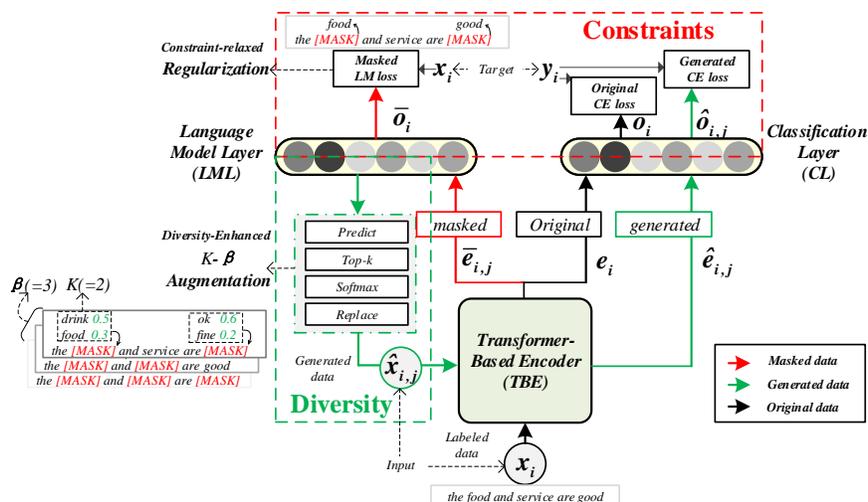
$$\hat{\mathbf{x}}_{i,j} = \phi(\mathbf{x}_i; k, \beta, \theta_a), j \in [1, \beta]. \quad (1)$$

Here,  $\phi$  is the operation in augmentation,  $\hat{\mathbf{x}}_{i,j} \in \mathbb{R}^{T \times V}$  is the  $j$ -th data generated based on  $\mathbf{x}_i$ ,  $\beta$  is the number of runs for data generation,  $\theta_a$  are the parameters of LM. The generated data  $\hat{\mathbf{x}}_{i,j}$  has the same label with  $\mathbf{x}_i$ .

The classifier learns the map function of

$$Y = f_{\theta_c}(\mathbf{X}), \quad (2)$$

where  $\{(\mathbf{X}, Y)\}$  is the joint of  $\{(\mathbf{x}_i, y_i)\}$  and  $\{(\hat{\mathbf{x}}_{i,j}, y_i)\}, j \in [1, \beta]$ .



**Fig. 2.** The structure of Diversity-Enhanced and Constraints-Relaxed Augmentation (DECRA).

## 4 Model Description

Fig. 2 shows the structure of the **Diversity-Enhanced and Constraints-Relaxed Augmentation (DECRA)**. Our DECRA has two essential components based on a backbone model which has a Transformer-Based Encoder (TBE), a Language

Model Layer (LML) and a Classification Layer (CL). Firstly, the  $k$ - $\beta$  augmentation is applied to the original data to generate diversity-enhanced data. Secondly, the masked Language Model (LM) loss is introduced as the regularization, which is the relaxed-constraint in generating.

#### 4.1 Transformer-based encoder

Transformer-Based Encoder (TBE) stacks multiple layers of transformers [34] to encode the text data into embeddings. It is initialized by a pre-trained Language Model (LM) which is trained on large-scale multi-domain datasets. The original data  $\mathbf{x}_i$  is masked into  $\bar{\mathbf{x}}_i$ . The original data  $\mathbf{x}_i$  is encoded as follows,

$$\mathbf{e}_i = \text{Transformer}_{\theta_t}(\mathbf{x}_i). \quad (3)$$

Here,  $\mathbf{e}_i \in \mathbb{R}^{T \times H}$  is the embeddings for classification,  $\text{Transformer}_{\theta_t}$  represents the processing of transformers,  $T$  is the length of original data,  $H$  is the hidden size of embeddings,  $\theta_t$  is the parameters of TBE. Similarly, we can get embeddings  $\bar{\mathbf{e}}_i$  for the masked data  $\bar{\mathbf{x}}_i$ .

#### 4.2 Language model layer

Language Model Layer (LML) is composed of a fully-connected layer. The fully-connected layer predicts the masked position based on its contextual embedding [8] that is fundamental for  $k$ - $\beta$  augmentation. It also essential to calculate the masked Language Model loss [8] on original data as a regularization. The  $\bar{\mathbf{e}}_i$  is embeddings of masked data. The prediction is calculated as,

$$\bar{p}_i = g_{\theta_a}(\bar{\mathbf{e}}_i). \quad (4)$$

Here,  $\bar{p}_i \in \mathbb{R}^{T \times V}$  represents the probabilities of tokens in masked positions,  $g_{\theta_a}(\cdot)$  maps the embedding size vector to vocabulary size.

#### 4.3 Classification layer

Classification Layer (CL) takes the first position of embeddings encoded by the TBE as input, and outputs the class categories. For labeled data, we calculate the predictions as follow,

$$o_i = f_{\theta_c}(\mathbf{e}_i), \quad (5)$$

where  $f_{\theta_c}(\cdot)$  represents the function of CL,  $\theta_c$  is the parameters of CL,  $o_i \in \mathbb{R}^C$  represents the predictions.

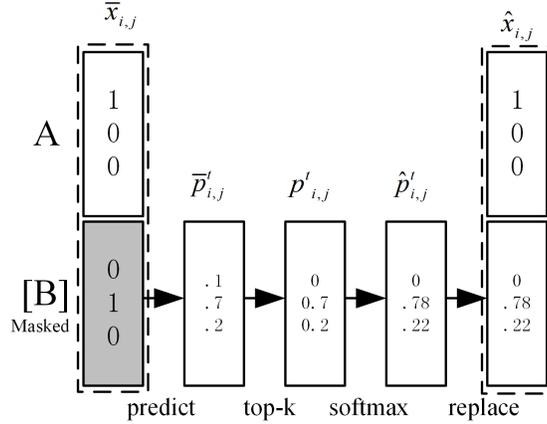
#### 4.4 $K$ - $\beta$ augmentation

$k$ - $\beta$  algorithm is designed to enhance the diversity in generating. It aims to augment the original data  $\mathbf{x}_i$   $\beta$  times to get the generated data  $\hat{\mathbf{x}}_{i,j}, j \in [1, \beta]$ .

$$\hat{\mathbf{x}}_{i,j} = \phi(\mathbf{x}_i; k, \beta, \theta_t, \theta_a), j \in [1, \beta]. \quad (6)$$

Here,  $\phi$  is the  $k$ - $\beta$  augmentation,  $\beta$  is the set of masks as well as the times of augmentation applied,  $k$  is the number of tokens used for replacing the masked position,  $\theta_t$  and  $\theta_a$  are the parameters in TBE and LML respectively. By this, the changing scope of generated data is expanded. Also, the degree of complexity of generated data is enhanced.

For each time of augmentation, the original data  $\mathbf{x}_i$  is randomly masked  $\bar{\mathbf{x}}_i$  for augmentation, as shown in Fig. 3. Data augmentation consists of four steps: predict, top-k, softmax and replace.



**Fig. 3.** The demonstration of  $k$ - $\beta$  augmentation.  $V = 3$ ,  $T = 2$  and  $k = 2$ . For a given sample  $\bar{\mathbf{x}}_{i,j}$ , the masked position is  $t = 2$ , the masked token is  $\mathbf{B}$ . The  $\bar{p}'_{i,j}$  is generated based on  $\bar{\mathbf{e}}_{i,j}$ , the  $p'_{i,j}$  is the top-k of  $\bar{p}'_{i,j}$ , the  $\hat{p}'_{i,j}$  is the normalized  $p'_{i,j}$ .

**Predict.** The embedding of the randomly masked position is feed into LML to get the predictions  $\bar{p}'_{i,j} \in \mathbb{R}^V$ . The predictions represent the probabilities of tokens to fit the  $t$ -th position.

**Top-k.** The top-k sampling, which is often used to improve the diversity in data augmentation [9], is used. The top-k probabilities tokens in  $\bar{p}'_{i,j}$  are selected as  $p'_{i,j} \in \mathbb{R}^k$ .

**Softmax.** The top-k probabilities are feed into a softmax function to normalize the probabilities.

$$\hat{p}'_{i,j} = \text{softmax}(p'_{i,j}) \quad (7)$$

Here,  $\hat{p}'_{i,j} \in \mathbb{R}^k$  is the normalized top-k probabilities.

**Replace.** For the convenient of replacement [15], we fill the value of  $\hat{p}'_{i,j}$  into a zero vector to get  $p^t_{i,j} \in \mathbb{R}^V$ . Instead of only one sampled token, we use  $k$  tokens to replace the masked token  $\bar{\mathbf{x}}_{i,j}$ , and get the generated data  $\hat{\mathbf{x}}_{i,j}$ . Note that the number of masked tokens is not fixed. The progress is repeated  $\beta$  times to get  $\hat{\mathbf{x}}_{i,j}, j \in [1, \beta]$ . The labels  $\hat{\mathbf{x}}_{i,j}$  all set to  $y_i$  as the setting in [37]. The

generated data are encoded for classification  $\hat{\mathbf{e}}_{i,j}, j \in [1, \beta]$  as in Eq. 3. Then, as in Eq. 5, we can get the prediction of generated data after  $k$ - $\beta$  augmentation  $\hat{o}_{i,j}, j \in [1, \beta]$ .

#### 4.5 Regularization

Masked Language Model (LM) loss [8] generates relaxed contextual constraints compared to fine-tuning. The labeled data is corrupted by randomly replacing some positions into maskers. Then, the model learns to predict the original token with the contextual embedding in the masked position. It takes the embeddings of the masked position  $\bar{\mathbf{e}}_i$  as inputs, takes the original tokens  $\mathbf{x}_i$  as labels, and calculates the loss as follow,

$$\mathcal{L}_{LM} = \frac{1}{M} \sum_{t=1}^T m_t \mathbf{x}_i^t \log(f_{\theta_a}(\bar{\mathbf{e}}_i^t)), \quad (8)$$

where  $\mathcal{L}_{LM}$  represents the masked LM loss,  $m_t = 1$  indicates the token on  $t$  position is masked,  $\mathbf{x}_i^t \in \mathbb{R}^V$  is the  $t$ -th token,  $\theta_a$  is the parameters of LML,  $M$  is the number of masked positions.

#### 4.6 Training process

As described in Algorithm 1, the cross-entropy between the predictions and  $y_i$  is calculated as

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^N y_i \log(o_i), \quad (9)$$

where  $N$  is the total number of original data.

Similarly, we can get cross-entropy loss  $\hat{\mathcal{L}}_{CE}$  for the data generated by  $k$ - $\beta$  augmentation,

$$\hat{\mathcal{L}}_{CE} = -\frac{1}{N} \frac{1}{\beta} \sum_{i=1}^N \sum_{j=1}^{\beta} y_i \log(\hat{o}_{i,j}). \quad (10)$$

Here, we average the loss calculated on  $\beta$  generated data which can get a more stable improvement [6].

The final loss is weighted average as follow,

$$\mathcal{L}_{final} = \mathcal{L}_{CE} + \lambda_a \hat{\mathcal{L}}_{CE} + \lambda_{lm} \mathcal{L}_{LM}. \quad (11)$$

Here, The  $\lambda_a$  and  $\lambda_{lm}$  are the weights for each loss term.

## 5 Experiments

### 5.1 Experimental settings

**Dataset** To evaluate the text augmentation in low-resource classification, we use the same settings in [15]. We evaluate the UABC model based on three

---

**Algorithm 1** The training algorithm of DECRA.

---

**Require:** corpus  $\{(\mathbf{x}_i, y_i)\}, i \in N, \lambda_a$  and  $\lambda_{lm}, \beta$  and  $k$ .  
Initialize  $\theta_a$  and  $\theta_t$  by a pre-trained BERT  
Initialize  $\theta_c$   
**for**  $epoch = 1, \dots, M$  **do**  
  **for**  $i = 1, \dots, N$  **do**  
    Masking  $\mathbf{x}_i$  to get  $\bar{\mathbf{x}}_i$   
    Get embeddings  $\mathbf{e}_i, \bar{\mathbf{e}}$  through  $Transformer_{\theta_t}$   
    Calculating the  $\mathcal{L}_{LM}$  based on  $g_{\theta_a}(\bar{\mathbf{e}}_i)$  and  $\mathbf{x}_i$   
    Getting the generated data  $\hat{\mathbf{x}}_{i,j}, j \in [1, \beta]$  through  $\phi(\mathbf{x}_i; k, \beta, \theta_t, \theta_a)$   
    Calculating the  $\hat{\mathcal{L}}_{CE}$  based on  $f_{\theta_c}(Transformer_{\theta_t}(\hat{\mathbf{x}}_{i,j}))$  and  $y_i$   
    Calculating the  $\mathcal{L}_{CE}$  based on  $f_{\theta_c}(\mathbf{e}_i)$  and  $y_i$   
     $\mathcal{L}_{final} = \mathcal{L}_{CE} + \lambda_a \hat{\mathcal{L}}_{CE} + \lambda_{lm} \mathcal{L}_{LM}$   
    Update the gradients of  $\theta_t, \theta_a$  and  $\theta_c$   
  **end for**  
**end for**

---

benchmark classification datasets, including TREC, SST-5, and IMDB. TREC is to categorize a question into six question types [19]. SST-5 is the Stanford Sentiment Treebank with five categories of very positive, positive, neutral, negative and very negative [30]. IMDB is for binary movie review sentiment [21], Table 1 summarizes the statistics of the three datasets. For each dataset, we randomly sample 15 small datasets. Each contains 40 samples per class for training and 5 (except SST-5 is 2) samples per class for validation. The models are evaluated on the validation set at the end of each epoch. The optimal model on the validation set is evaluated on the full-size testing set. The mean accuracy of the 15 small datasets used as the final result to evaluate the model performance on each dataset. The average of the mean accuracy on three datasets is the overall score for each model.

**Table 1.** The statistics of datasets.  $c$ : Number of target classes.  $l$ : Average sentence length.  $Train$ : Train set size.  $Val$ : Validation set size.  $Test$ : Test set size.

Data	$c$	$l$	$Train$	$Val$	$Test$
SST-5	5	19	200	10	2210
IMDB	2	252	80	10	2500
TREC	6	10	240	30	500

## 5.2 Comparison methods

We compare our model with six methods that can be utilized for Low-Resource Classification (LRC). BERT (base, uncased) for text classification without augmentation [8] is the *baseline*. Five augmentation methods are listed as follow:

- **EDA** [36] is a recent data augmentation approach containing a set of four text augmentation techniques, including synonym replacement, random insertion, random swap, and random deletion.
- **BT**<sup>1</sup> [29] translates the labeled data into another language and then translates it back into the original language.
- **Mixup** [16] generates out-of-manifold samples through linearly interpolating data representations and their corresponding labels of random sample pairs.
- **CBERT** [37] is the latest model-based augmentation that uses a conditional BERT, which is pre-trained on a training set, for augmentation.
- **LDMAW** [15] is the state-of-the-art augmentation that uses reinforcement learning to train both the augmentser BERT and classifier BERT for LRC.

### 5.3 Implementation details

We use BERT-base [8] to initialize our transformer-based encoder and language model layer, and randomly initialize the classification layer. We use Adam optimization [17] with an initial learning rate of  $2e - 5$ . The epoch is set to 20 and the batch size is 8 for all datasets. For each minibatch data, we use  $k$ - $\beta$  augmentation with  $\beta = 18$  and  $k = 2$ . The weights of losses are  $w_a = 1$  and  $w_{lm} = 1.5$ . For each experiment, the model is evaluated on the validation set after every training epoch, and the optimal epoch on the validation set is evaluated on the test set.

### 5.4 Classification in low-resource condition

**Table 2.** DA extrinsic evaluation in low-resource settings. Results are reported as Mean (STD) accuracy on full test set. Experiments are repeated 15 times. <sup>†</sup> refers to the results reported in [15]

Methods	Datasets			AVG
	SST5(200)	IMDB(80)	TREC(240)	
Baseline <sup>†</sup> [8]	33.3±6.2	63.6±4.4	88.3±2.9	61.7
EDA [36]	36.8±6.1	62.8±6.0	86.6±4.1	62.1
BT [29]	35.8±4.3	66.4±4.2	86.6±4.3	62.8
Mixup [16]	36.0±4.0	67.3±5.1	88.3±3.2	63.9
CBERT <sup>†</sup> [37]	34.8±6.9	63.7±4.8	88.3±1.1	62.3
LDMAW <sup>†</sup> [15]	37.0±3.0	65.6±3.7	89.2±2.1	63.9
<b>DECRA (our work)</b>	<b>40.3±3.4</b>	<b>69.0±4.0</b>	<b>89.5±1.6</b>	<b>66.3</b>

Table 2 exhibits the results of all models on three datasets. Our DECRA outperforms all baselines on all three datasets. Firstly, our DECRA can improve the

<sup>1</sup> We implement the back translation based on MarianMT in Transformers, <https://huggingface.co/transformers>.

classification performance in LRC from 63.9% to 66.3%. When compared with LDMAW and Mixup, our model achieves a higher overall score. That benefits from the effects of our  $k$ - $\beta$  augmentation which effectively enhances the diversity of generated data. Secondly, our DECRA achieves the highest mean accuracy score on every dataset. The stable improvement may benefit from the expanded changing scope in  $k$ - $\beta$  augmentation. Thirdly, our DECRA has a smaller parameters-scale than LM-based approaches. When compared with CBERT and LDMAW, our model unifies the augments and classifier by reducing nearly half of the parameters. Noticeable that the LDMAW uses reinforcement learning to tune the augments(BERT) for the classifier(BERT). Our DECRA improves the overall score by a significant margin. It’s 3.8% improvements against the LDMAW and 6.3% improvements against CBERT. The improvement benefits from the improvement of generalization ability.

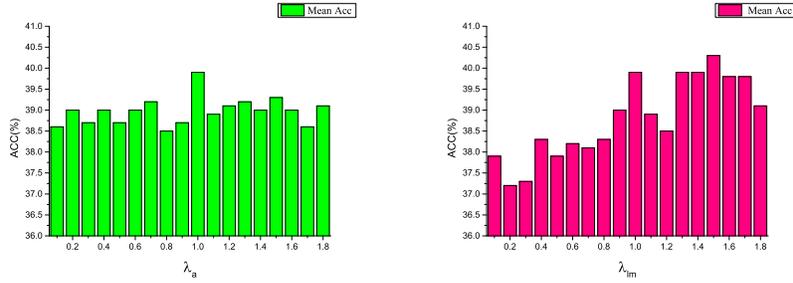
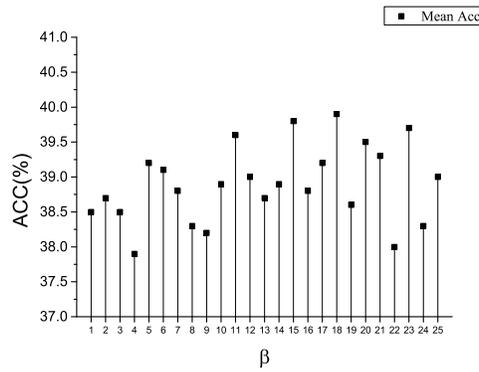
### 5.5 Ablation study

**Table 3.** The ablation results(%) of DECRA model. Results are reported as Mean (STD) accuracy on full test set. Experiments are repeated 15 times.

ID	$k$ - $\beta$	Reg.	Dataset			AVG
			SST5(200)	IMDB(80)	TREC(240)	
1	×	×	33.3±6.2	63.6±4.4	88.3±2.9	61.7
2	×	✓	33.8±2.9	64.6±4.4	86.5±3.4	61.6
3	✓	×	36.5±3.2	65.6±5.0	89.0±1.8	63.7
4	×	△	38.2±3.3	68.8±4.1	88.4±3.0	65.1
5	✓	△	39.0±5.1	68.7±5.4	88.7±1.9	65.5
6	✓	✓	40.3±3.4	69.0±4.0	89.5±1.6	66.3

× indicates the component is removed from DECRA, ✓ indicates the component is added in DECRA.  $k$ - $\beta$  is the  $k$ - $\beta$  augmentation and Reg. is the masked LM loss. △ indicates the DECRA is pre-trained with masked LM loss and then finetuned as [12].

To better understand the working mechanism of the DECRA, we conduct ablation studies on all three datasets, as listed in Table 3. 1) Without augmentation, the  $ID2$ , which has relaxed constraints compared to  $ID4$ , results in lower classification accuracy. The results indicate that strong constraints are more effective in LRC without augmentation. 2) With augmentation, the  $ID6$ , which has relaxed-constraints compared to  $ID5$ , promotes the overall score from 65.5 to 66.3. The improvement of the overall score in LRC with augmentation mainly from the relaxed-constraints. 3) Besides, the  $ID3$  outperforms  $ID1$  due to the diversity-enhanced  $k$ - $\beta$  augmentation. Also, the  $ID5$  has a higher overall score (65.5) than  $ID4$  (65.1). The results show the effects of  $k$ - $\beta$  augmentation in LRC, which enhance the diversity of generated data.

(a) Results of different setting of  $\lambda_a$ . (b) Results of different setting of  $\lambda_{lm}$ .**Fig. 4.** The results of different loss weights on SST5.**Fig. 5.** The results of different  $\beta$  on SST-5..

## 5.6 Importance of diversity and constraints.

To analyze the importance of diversity and constraints ( $\tilde{\mathcal{L}}_{CE}$  and  $\mathcal{L}_{LM}$ ), we grid search the optimal weights ( $\lambda_a$  and  $\lambda_{lm}$ ) on SST5. Experiments are repeated 15 times. Firstly, the  $\lambda_{lm}$  is set to 1.0 in the searching of the  $\lambda_a$ . Then, the  $\lambda_a$  is set to the optimal (1.0) in the search of  $\lambda_{lm}$ . Fig. 4(a) describes the effects of weight  $\lambda_a$  for  $k$ - $\beta$  augmentation  $\tilde{\mathcal{L}}_{CE}$ . The average accuracy achieves the peak when the  $\lambda_a$  is 1.0. The generated data has equal importance to the original data. This setting is identical to [15]. Fig. 4(b) shows the effects of weight  $\lambda_a$  for masked LM loss  $\mathcal{L}_{LM}$ . The model reaches the optimal classification performance when the  $\lambda_{lm}$  is 1.5. The  $\mathcal{L}_{LM}$  has larger weights than  $\mathcal{L}_{CE}$ . It shows the importance of contextual constraints in LRC.

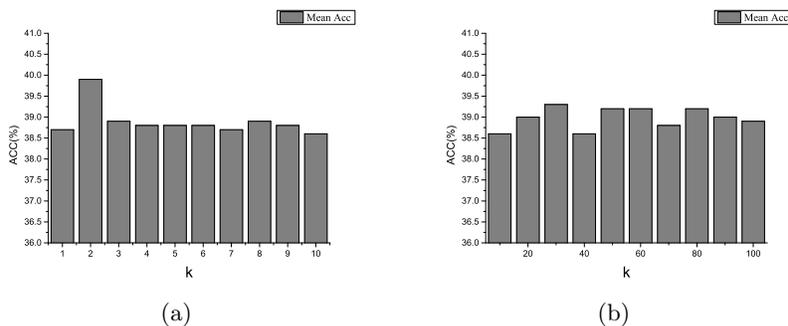


Fig. 6. The results of different  $k$  settings on SST-5.

### 5.7 $k$ - $\beta$ augmentation on enhancing diversity

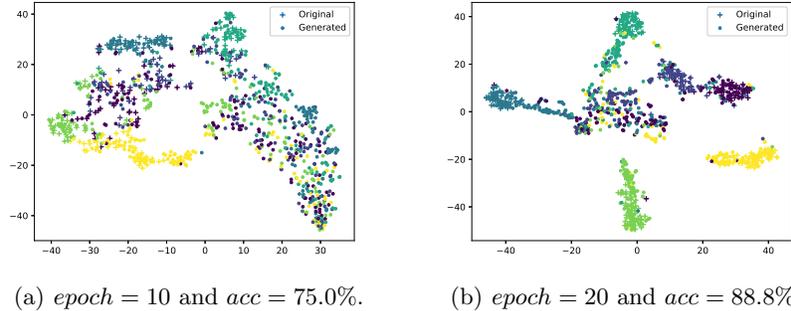
To analyze the effects of the hyperparameter in  $k$ - $\beta$  augmentation, we conduct two groups of experiments. The  $\lambda_a$  and  $\lambda_{lm}$  are set to 1.0 as default. Experiments are repeated 15 times.

**Degree of complexity** To explore the effects of  $k$  (degree of complexity) in  $k$ - $\beta$  augmentation, we conduct two sets of experiments. As shown in Fig. 6, one set is  $k \in [1, 2, \dots, 10]$  and another is  $k \in [10, 20, \dots, 100]$ .  $k$  is the number of tokens to replace the masked token. In LRC, the operation increases the diversity in generating. Through the top- $k$  operation, the generated data will contain complex information from  $(\text{number of masks})^k$  samples generated by the previous operation in [37]. As shown in Fig. 6(a), the model achieves the highest overall score when  $k = 2$ . The mean accuracy decreases along with an increase of  $k$ . That may be caused by noise information. The model achieves a lower mean accuracy when  $k = 1$ . That is expected as a result of lacking diversity in generating. The results indicate that  $k = 2$  is an optimal choice for DECRA.

**Changing scope** To explore the effect of  $\beta$  (changing scope) in  $k$ - $\beta$  augmentation, we evaluate the model under the setting of  $\beta \in [1, 25]$  and  $k = 2$ . The  $\beta$  is the times of generating maskers for original data. As shown in Fig. 5, as  $\beta$  goes from 1 to 18, the mean accuracy of the model tends to increase generally. This benefits from enhancing the changing scope of the augmentation. When  $\beta = 18$ , the highest mean accuracy is achieved for generated diverse data for classification. As the increase of  $\beta$  from 18 to 25, the model performance begins to fluctuate around 39%. The  $\beta$  reaches its limit in improving diversity. Therefore, we choose 18 as the optimal setting.

### 5.8 Visualization

To present the effectiveness of the diversity and constraints in DECRA, we visualized the labeled data and generated data in the subset 0 of TREC with the



**Fig. 7.** The visualization of original data and generated data on TREC ( $subset = 0$ ). Different colors are used to mark the original and generated data in each category.

settings,  $k = 2$ ,  $\beta = 3$ ,  $\lambda_a = 1.0$  and  $\lambda_{lm} = 1.0$ . The visualization data in two specific epochs,  $epoch = 10$  and  $epoch = 20$ , with TSNE [22] represents two different phrases in the training process. As shown in Fig. 7, we can observe the diversity of the generated data in all training phrases as well as the constraints. The generated data partly distributes around the cluster of its “should be” class and partly distributes distantly away. That demonstrates the effects of diversity and constraints in DECRA. It proves our DECRA works well in LRC.

## 6 Conclusion

In Low-Resource Classification (LRC), the currently used augmentation approaches, such as LDMAW, suffer from generalizing poorly due to strong constraints but weak diversity. To address this dilemma, we propose a Diversity-Enhanced and Constraints-Relaxed Augmentation (DECRA). The DECRA has two essential components on top of a transformer-based backbone model. We propose a  $k$ - $\beta$  augmentation to enhance the diversity of generated data by expanding the changing scope and enhancing the degree of complexity in generated data. We introduce the masked Language Model loss instead of staged fine-tuning to generate relaxed-constraints. The improved diversity and relaxed constraints help to generate data scattered near or approach the category boundaries. Trained with both labeled and generated data in low-resource conditions, the model achieves better generalization ability. Experimental results demonstrate that our DECRA significantly outperforms state-of-the-art augmentation techniques in low-resource classification. The results may shed some light on LRC.

## References

1. Abu-Nasser, B.: Medical expert systems survey. International Journal of Engineering and Information Systems (IJEAIS) **1**(7), 218–224 (2017)

2. Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., Tepper, N., Zwerdling, N.: Do not have enough data? deep learning to the rescue! In: AAAI. pp. 7383–7390 (2020)
3. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 (2017)
4. Archambault, G.P., Mao, Y., Guo, H., Zhang, R.: Mixup as directional adversarial training. arXiv preprint arXiv:1906.06875 (2019)
5. Bertero, D., Siddique, F.B., Wu, C.S., Wan, Y., Chan, R.H.Y., Fung, P.: Real-time speech emotion and sentiment recognition for interactive dialogue systems. In: Proceedings of the 2016 conference on empirical methods in natural language processing. pp. 1042–1047 (2016)
6. Berthelot, D., Carlini, N., Cubuk, E.D., Kurakin, A., Sohn, K., Zhang, H., Raffel, C.: Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In: International Conference on Learning Representations (2019)
7. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 113–123 (2019)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1) (2019)
9. Fadaee, M., Bisazza, A., Monz, C.: Data augmentation for low-resource neural machine translation. arXiv preprint arXiv:1705.00440 (2017)
10. Guo, H.: Nonlinear mixup: Out-of-manifold data augmentation for text classification. In: AAAI. pp. 4044–4051 (2020)
11. Gupta, R.: Data augmentation for low resource sentiment analysis using generative adversarial networks. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7380–7384. IEEE (2019)
12. Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., Smith, N.A.: Don’t stop pretraining: Adapt language models to domains and tasks. arXiv preprint arXiv:2004.10964 (2020)
13. Hernández-García, A., König, P.: Data augmentation instead of explicit regularization. arXiv preprint arXiv:1806.03852 (2018)
14. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 (2018)
15. Hu, Z., Tan, B., Salakhutdinov, R.R., Mitchell, T.M., Xing, E.P.: Learning data manipulation for augmentation and weighting. In: Advances in Neural Information Processing Systems. pp. 15764–15775 (2019)
16. Jindal, A., Gnaneshwar, D., Sawhney, R., Shah, R.R.: Leveraging bert with mixup for sentence classification (student abstract). In: AAAI. pp. 13829–13830 (2020)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Kobayashi, S.: Contextual augmentation: Data augmentation by words with paradigmatic relations. arXiv preprint arXiv:1805.06201 (2018)
19. Li, X., Roth, D.: Learning question classifiers. In: COLING 2002: The 19th International Conference on Computational Linguistics (2002)
20. Li, Z., Liu, X., Zhang, G., Xie, N., Wang, S.: A multi-granulation decision-theoretic rough set method for distributed fc-decision information systems: An application in medical diagnosis. *Applied Soft Computing* **56**, 233–244 (2017)
21. Maas, A., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the

- association for computational linguistics: Human language technologies. pp. 142–150 (2011)
22. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. *Journal of machine learning research* **9**(Nov), 2579–2605 (2008)
  23. Marivate, V., Sefara, T.: Improving short text classification through global augmentation methods. In: *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. pp. 385–399. Springer (2020)
  24. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
  25. Miyato, T., Dai, A.M., Goodfellow, I.: Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725* (2016)
  26. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pp. 1532–1543 (2014)
  27. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. *OpenAI Blog* **1**(8), 9 (2019)
  28. Sato, M., Suzuki, J., Shindo, H., Matsumoto, Y.: Interpretable adversarial perturbation in input embedding space for text. *arXiv preprint arXiv:1805.02917* (2018)
  29. Shleifer, S.: Low resource text classification with ulmfit and backtranslation. *arXiv preprint arXiv:1903.09244* (2019)
  30. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*. pp. 1631–1642 (2013)
  31. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 2818–2826 (2016)
  32. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013)
  33. Tu, E., Yang, J.: A review of semi supervised learning theories and recent advances. *arXiv preprint arXiv:1905.11590* (2019)
  34. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: *Advances in neural information processing systems*. pp. 5998–6008 (2017)
  35. Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)* **53**(3), 1–34 (2020)
  36. Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196* (2019)
  37. Wu, X., Lv, S., Zang, L., Han, J., Hu, S.: Conditional bert contextual augmentation. In: *International Conference on Computational Science*. pp. 84–95. Springer (2019)
  38. Xia, M., Kong, X., Anastasopoulos, A., Neubig, G.: Generalized data augmentation for low-resource translation. *arXiv preprint arXiv:1906.03785* (2019)
  39. Young, T., Pandealea, V., Poria, S., Cambria, E.: Dialogue systems with audio context. *Neurocomputing* (2020)
  40. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017)
  41. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: *AAAI*. pp. 13001–13008 (2020)