Global Mixup: Eliminating Ambiguity in text classification with Clustering

Xiangjin Xie[1], **Yangning Li**[1,3], **Wang Chen**[2], **Kai Ouyang**[1],**Zuotong Xie**[1], **Hai-Tao Zheng**[1,3]

[1]Shenzhen International Graduate School, Tsinghua University
[2]Google Inc.
[3]Pengcheng Laboratory
[1]{xxj20,liyn20, oyk20,xzt20,hn20}@mails.tsinghua.edu.cn,
[2]allencw@google.com, [1]zheng.haitao@sz.tsinghua.edu.cn

Abstract

Data augmentation with **Mixup** has been proven an effective method to regularize the current deep neural networks. Mixup generates virtual samples and corresponding labels at once through linear interpolation. However, this one-stage generation paradigm and the use of linear interpolation have the following two defects: (1) The label of the generated sample is directly combined from the labels of the original sample pairs without reasonable judgment, which makes the labels likely to be ambiguous. (2) linear combination significantly limits the sampling space for generating samples. To tackle these problems, we propose a novel and effective augmentation method based on global clustering relationships named **Global Mixup**. Specifically, we transform the previous one-stage augmentation process into two-stage, decoupling the process of generating virtual samples from the labeling. And for the labels of the generated samples, relabeling is performed based on clustering by calculating the global relationships of the generated samples. In addition, we are no longer limited to linear relationships and therefore can generate more reliable virtual samples in a larger sampling space. Extensive experiments for **CNN**, **LSTM**, and **BERT** on five tasks show that Global Mixup significantly outperforms previous baselines. Further experiments also demonstrate the advantage of Global Mixup in low-resource scenarios.

Introduction

Although deep neural networks have achieved great results in computer vision (Krizhevsky, Sutskever, and Hinton 2012), speech recognition (Cui, Goel, and Kingsbury 2015) and Natural Language Processing (NLP), they are error-prone and poorly generalized when lacking sufficient training data. Data augmentation can effectively alleviate this problem by generating new samples transformed from the training sets. In NLP, the data augmentations methods consist mainly of rule-based and generation-based. Rule-based methods usually rely on manually designed paradigms, such as synonym replacement (Zhang, Zhao, and LeCun 2015), random noise injection, deletion, and insert (Wei and Zou 2019). Generation-based methods leverage trained deep models such as back-translation and pre-trained models to generate new samples automatically. Recently proposed Mixup (Zhang et al. 2018), and its variants (Zhang, Yu, and Zhang 2020; Verma et al. 2019; Guo, Mao, and Zhang 2019) further improve the efficiency and robustness of models by features linear interpolation, which generate more virtual samples in the feature space.

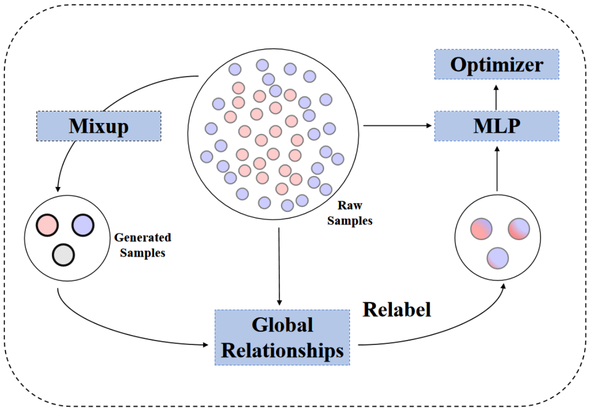

Figure 1: Illustration of the two-stage paradigm of Global Mixup, a new paradigm for data augmentation.

While achieving promising results, Mixup still has limitations. First, the labels of the generated samples with data augmentation were directly inherited from the labels of the original samples, and the gaps between the generated sample features and the original samples were not adequately represented, which have lead to errors in the labels of the generated samples. Second, The restriction of the data generated by Mixup is within the linearity and only two samples' similarity is considered, which results in the ambiguity of Mixup .

To address these problems, we propose **Global Mixup**, a data augmentation method that eliminates ambiguity by decoupling the sample generation phase from the label determination phase and relabeling by clustering relationships of samples. Specifically, Global Mixup separates the sample generation and labels determination for data augmentation into two stages, and relabels potentially ambiguous labels by clustering. The generated samples will not directly inherit the labels of the original samples or sample pairs. For the ambiguous labeling problem of the generated samples, Global Mixup labels the generated sample based on its global relationships with the original set of samples, so that the labels of the generated samples reflect the clustering relationship with the original samples. Thus, the generated samples are uniquely labeled through global relationships, mitigating the ambiguity inherent in Mixup, which only considers local linear relationships. Then, for the distribution problem of the generated samples, because Global Mixup's sample generation and labeling processes are separate, the generated data of Global Mixup can be obtained from broader distributions to scale the training data more efficiently. **Figure** 1 shows the process of Global Mixup, it's a new paradigm for data augmentation, through split sample generation and label determination, the generated samples will get more accurate labels which can reduce the error optimization during model training.

Experiments on the classical models and pre-trained models show that Global Mixup significantly outperforms the rule-based methods and Mixup (Guo, Mao, and Zhang 2019) on different text classification tasks. The advantage of this method is more evident in low-resource scenarios, using 23% of the training data on SST-1 and 36% of the training data on TREC exceeds the accuracy of baseline with all training data.

In a nutshell, our main contributions are three-fold:

(1) To the best of our knowledge, we were the first to split sample generation and label determination into two separate phases in augmentation and obtain more accurate labels for the generated samples based on clustering relationships.

(2) We present a novel data augmentation approach termed Global Mixup, which implies stable labels to the virtual samples and avoids the emergence of ambiguous, overconfident labels in linear interpolation methods. Moreover, theoretically, because of the separation of the sample generation and labeling processes, Global Mixup is capable of labeling arbitrary samples, not limited to those inside convex combinations.

(3) Extensive experiments on five datasets and three models (including pre-trained models) demonstrate the effectiveness of Global Mixup, especially in few-shot scenarios.

Related Work

Data augmentation has become a prevalent research topic in recent years to solve the data scarcity problem. Automatic data augmentation has improved significant performance on various tasks such as computer vision (Simard et al. 1998; **?**) and speech tasks (Cui, Goel, and Kingsbury 2015). However, only rare research exploits data augmentation in natural language processing tasks because of the high complexity of language and words' discreteness. Dominant data augmentation and Interpolation-based data augmentation are two main kinds of methods that can be introduced into NLP tasks.

Dominant data augmentation

The dominant data augmentation approach focuses on generating new sentences similar to the labeled data by introducing external knowledge:

Rule-based data augmentation Rule-based methods generate samples by transforming the original sample with human-designed rules, such as (Wei and Zou 2019) using synonym substitution, random insertion, random exchange, and random deletion. (Zhang, Zhao, and LeCun 2015) replace words based on an English thesaurus. (Coulombe 2018) proposes synonymous substitution, according to the types of words suitable for replacement: adverbs, adjectives, nouns, verbs, and simple pattern matching conversion and grammar tree conversion using regular expressions to generate new sentences. Other works (Wang and Yang 2015) also try to use the most similar words for text replacement based on pre-trained word vectors such as Glove(Pennington, Socher, and Manning 2014), Word2vec(Mikolov et al. 2013).

Generation-based data augmentation Generation-based methods focus on generating sentences based on language models. (Sennrich, Haddow, and Birch 2016) utilize an automatic back-translation to pair monolingual training data as additional parallel training data. (Kober et al. 2021) use generative adversarial networks (GANs) (Goodfellow et al. 2014) to generate new training examples from existing ones. (Yu et al. 2018) consider back-translation based on a neural machine translation model. (Xie et al. 2019) introduces data noise in neural network language models. Recently, pre-trained language models are also used to generate new labeled data based on contextual information (Kobayashi 2018). (Wu et al. 2019) apply the conditional BERT (Kenton and Toutanova 2019) model to enhance contextual augmentation. However, the data generated by dominant data augmentation methods are similar to the original data, leading to the model still learning similar patterns. Therefore, the model cannot handle data scarcity problems when the test data distribution differs from the training data.

Interpolation-based data augmentation

Interpolation-based data augmentation has been proposed in Mixup (Zhang et al. 2018). Mixup extends the training data by training a neural network on convex combinations of pairs of examples and their labels, as shown in Preliminaries . Mixup has achieved relative success in many computer vision tasks. Mixup variants (Verma et al. 2019; Summers and Dinneen 2019) use interpolation in the hidden representation to capture higher-level information and obtain smoother decision boundaries. Recently, more researchers have focused on utilizing Mixup to improve the model's performance in NLP tasks. wordMixup (Guo, Mao, and Zhang 2019) first performs interpolation on word embeddings and sentence embeddings, this method demonstrates the effectiveness of data augmentation methods that do not generate real sentences. SeqMix (Zhang, Yu, and Zhang 2020) generates subsequences along with their labels by using linear interpolation. These methods optimize Mixup by modifying the data generation based on Mixup and have proven effective. However, linear interpolation methods only take the relationships between two samples for the labels.

Methodology

We present overviews of the method composition of Global Mixup in **Figure** 2. The purpose of Global Mixup is to separate the sample generation and label determination process of data augmentation and to obtain accurate samples' labels by the similarity of samples, and encouraging the models to focus on the clustering relationships of samples to resolve the ambiguity of linear interpolation. To achieve this, we inherit the way Mixup generates virtual samples and change the way it labels samples.

Preliminaries

We first briefly describe the original Mixup (Zhang et al. 2018) and the variant of Mixup for text classification, wordMixup (Guo, Mao, and Zhang 2019).

Mixup: (Zhang et al. 2018) is the first data augmentation method proposed for image classification tasks that implements linear interpolations to mix different images and their labels to generate new samples in order to train models to recognize image features and classification in complex situations, it is similar to an image noise. In short, let (x, y) denote a sample of training data, where x is the raw input samples and y represents the one-hot label of x, the Mixup generates virtual training samples $(\tilde{x}, \tilde{y})$ can be formulated as follows:

$$\begin{aligned}\tilde{x} &= \lambda x_i + (1-\lambda)x_j,\\ \tilde{y} &= \lambda y_i + (1-\lambda)y_j,\end{aligned} \tag{1}$$

where (x_i, y_i) and (x_j, y_j) are two original samples drawn at random from training data, the mixing coefficient $\lambda \sim Beta(\alpha, \alpha)$, for $\alpha \in (0, \infty)$, and $Beta$ means the Beta distribution. Unlike the original sample, which uses hard labels, the generated virtual data uses soft labels. Then both the generated virtual samples and the original samples are used to train the network. **wordMixup**: (Guo, Mao, and Zhang 2019) is a linear interpolation method for text classification. Firstly, it converts all sentences into embedding matrix and pads them to the same length. For a set of training texts, they will all be represented as the same dimensional matrix $B \in R^{N \times d}$, where N represents the length of each text after padding and d represents the dimension of the vector for each word. Secondly, (B_i, y_i) and (B_j, y_j) are drawn at random from original train set, where y_i and y_j denote the corresponding class label of the sentence using one-hot representation. In short, the process of virtual training sample $(\widetilde{B}, \widetilde{y})$ generated by wordMixup can be formulated as follows:

$$\begin{aligned}\widetilde{B} &= \lambda B_i + (1-\lambda)B_j,\\ \widetilde{y} &= \lambda y_i + (1-\lambda)y_j,\end{aligned} \tag{2}$$

where the mixing coefficient $\lambda \sim Beta(\alpha, \alpha)$ is the same as in the Mixup, and α is set as 1 in wordMixup. For a deep neural network $f_k(x)$, cross-entropy loss L is used by wordMixup:

$$L = \widetilde{y} log(\mathrm{softmax}(W f_k(\widetilde{B}))), \tag{3}$$

where $W \in R^{c \times m}$ and $f_k(x)$ generate the m-dimensional vector.

Global Mixup

In **Vanilla Mixup**, including Mixup and the variations of Mixup, the generated virtual samples may have label ambiguity problems in the regions where linear interpolation of randomly selected original samples are intersections. For example, the Mixup aims to generate a virtual sample by linear interpolation as shown in **Figure** 2, but the same virtual sample which comes from different pairs of original samples will receive different labels as shown in the **Figure** 2 (a) and (b). And as shown in**Figure** 2 (b), when extremely different sample pairs are selected for mixup and intersection occurs, virtual samples may be generated that are similar but with opposite and overconfident labels. We call this phenomenon that the label gap between similar virtual samples generated based on different sample pairs is too large, the label ambiguity problem. To tackle the label ambiguity problem, we propose to calculate the global relationships of the generated virtual samples. Specifically, as shown in **Figure** 2 (b), When we generate the same virtual sample C based on two sample pairs (A_1, A_2) and (B_1, B_2) that have different labels, if we are using **Vanilla Mixup**, there will be a conflict in labeling the virtual sample C because the sample pair (A_1, A_2) corresponds to a different label than (B_1, B_2). Moreover, when the distribution of generated virtual samples is similar, ambiguity phenomenon often occurs. But, as shown in the figure 2(c), for the generated virtual samples G, the label is generated by computing the global relationship of G with all the original training samples using **Global Mixup**. Thus it will get a globally unique label, thus eliminating the ambiguity. Also, labeling and generation are independent when using Global Mixup, the generated samples can not be limited to the linear relationships of the original samples, which provides more options for generated samples in the distribution space. Specifically, training the neural networks using Global Mixup mainly consists of the following four steps:**Raw Samples Selection**: In this step, we randomly select a part of the sample pairs (B_i, y_i) and (B_j, y_j) from training data as raw materials for generating virtual samples. **Raw Mixed Samples Generation**: After randomly selecting the raw samples, we perform linear interpolation on them and generate virtual training samples$(\widetilde{B}, \tilde{y})$ as shown in Equation 2. For simplicity, the Vanilla Mixup samples generation method is used here. **Labels Reconfiguration**: In this part, we select a part of raw mixed sample for relabeling, usually choosing those raw mixed samples with overconfident labels. Specifically, we select samples with label $\tilde{y}$ satisfying $\arg\max \tilde{y} \geq \theta$ from the generated virtual sample set for relabeling, which means that the labels of overconfident virtual samples will be recalculated. The selection parameter $\theta \in [1/c, 1]$, c is the number of target labels. When $\theta = 1/c$, all raw mixed samples will be selected for relabeling. and when $\theta = 1$, it reduces to the Vanilla Mixup principle. Reconstruction of the labels of these virtual samples is as follows:

$$y^{\star} = \sum_{t=1}^{T} P(B_t \mid D(B_t, B'))y_t, \tag{4}$$

where $y^{\star}$ is the new label for B'. $P(B_t \mid D(B_t, B'))$ is the weight of y_t to generate $y^{\star}$, and $D(B_t, B')$ is the equation for computing the relationships between the training samples B_t and the generated virtual sample B'. It can be formalized as follows:

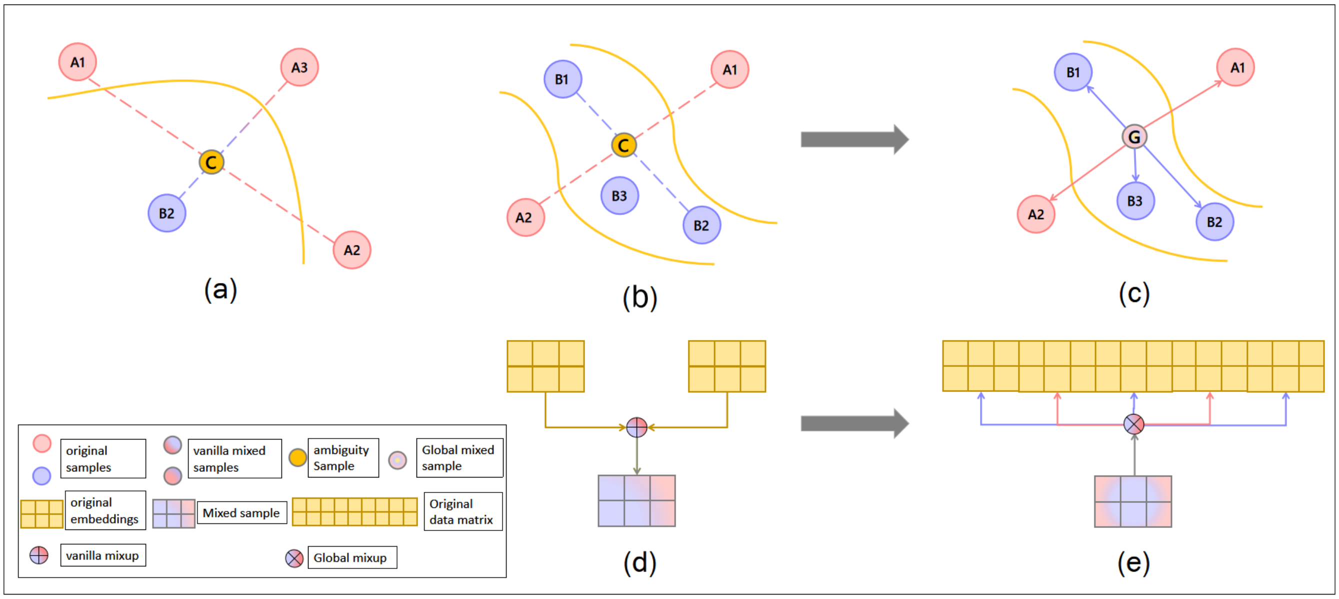

Figure 2: An overview of Global Mixup: (a) and (b) represent ambiguous cases in Vanilla Mixup, and (c) represents the Global Mixup with relabeling of the ambiguous samples in (b). (d) represents the way (a) and (b) generate samples and labels, and (e) represents the way (c) relabels the samples.

$$P(B_t \mid D(B_t, B')) = \frac{\exp(D(B_t, B'))}{\sum_{i=1}^{T} \exp(D(B', B_i))}, \tag{5}$$

where T is the total number of samples used to calculate the global relationships, the largest $top - s$ from all computed D will be used for the computation of P, and P of $B_w(w \in T - s)$ will be set to 0. $s \in [2, N]$ is the number of samples referenced to calculate the global relation of B'. When $s = 2$, only the relationships between B' and the samples that generate it will be calculated. When $s = N$ all training samples are calculated, and in general, We choose T equal to the number of batch size.

$$D(B^t, B') = \frac{\gamma \sum_{i=1}^{N} \sum_{j=1}^{d} B^t_{i,j} \cdot B'_{i,j}}{\sum_{i=1}^{N} \sum_{j=1}^{d} \sqrt{B^t_{i,j} \cdot B^t_{i,j}} * \sqrt{B'_{i,j} \cdot B'_{i,j}} + \epsilon}, \tag{6}$$

where D means the similarity of the matrices B^t and B'. And D can be interpreted as flattening the word embedding matrices B^t and B' and computing the cosine similarity of the two flattened matrices. Where d, N are the dimensional parameters of the matrix B. γ is the parameter of relationships correction, and ϵ is the parameter to prevent the denominator from being 0. For the **BERT** model, due to the attention mask mechanism, we change the formula for calculating D as follows, $A \in R^{1*N}$ represents the attention mask vector for each sentence:

$$D(B_t, B') = \frac{\gamma A_t B_t (A' B')^T}{\sqrt{A_t B_t (A_t B_t)^T * A' B' (A' B')^T + \epsilon}}. \tag{7}$$

Network Training: Finally, we use the original samples (B, y), virtual samples $(\tilde{B}, \tilde{y})$ generated by vanilla mixup and $(B', y^\star)$ generated by Global Mixup to train the network, compute the loss value and gradients update the parameters of the neural networks.

Mathematically, Global Mixup minimizes the average of the loss function L, The loss is combined three parts of loss:

$$\begin{aligned} \mathrm{L(f)} =& \delta \ell_{\text{orig}} + \tau \ell_{\text{vanilla}} + \eta \ell_{\text{global}} \\ =& \underset{(B,y)\sim P}{E} \; \underset{(\tilde{B},\tilde{y})\sim P}{E} \; \underset{(B',y^\star)\sim P}{E} \; \underset{\lambda\sim \mathrm{Beta}(\alpha,\alpha)}{E} \\ & \ell(f_k(\mathrm{Mix}_\lambda(B, \tilde{B}, B'), \mathrm{Mix}_\lambda(y, \tilde{y}, y^\star)). \end{aligned} \tag{8}$$

where $(B, y) \sim P$ represents the original distribution of the training data; $(\tilde{B}, \tilde{y}) \sim P$ represents the original distribution of the virtual data which generated by Vanilla Mixup; $(B', y^\star) \sim P$ represents the original distribution of the virtual data which generated by Global Mixup. P is the same as mixup (Zhang et al. 2018), represents the distribution of samples. ℓ represents the loss function. $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$ represent the distribution of λ. δ, τ, η are discount factors and f_k is the network to be trained.

Advantages of computing global relationships

Global Mixup separates sample generation and label generation into two stages and relabeling by computing global relationships, so it eliminates the ambiguity problem in Mixup and the variations of Mixup. And due to the separateness of sample generation and label generation it can label any sample, not limited to convex combinations consisting of training sets like linear interpolation, which allows Global Mixup to be used together with other generation methods and assist them in labeling.

Experiments

We conduct experiments on five tasks and three networks architectures to evaluate the effectiveness of Global Mixup.

Datasets

We conduct experiments on five benchmark text classification tasks and table 1 summarizes the statistical characteristics of the five data sets:

1. **YELP**: (yelp 2015), which is a subset of Yelp's businesses, reviews, and user data.
2. **SUBJ**: (Pang and Lee 2004), which aims to classify the sentences as subjectivity or objectivity.
3. **TREC**: (Li and Roth 2002), is a question dataset with the aim of categorizing a question into six question types.
4. **SST-1**: (Socher et al. 2013), is Stanford Sentiment Treebank, five categories of very positive, positive, neutral, negative, and very negative, Data comes from movie reviews and emotional annotations.
5. **SST-2**: (Socher et al. 2013), is the same as SST-1 but with neutral reviews removed and binary labels, Data comes from movie reviews and emotional annotations.

Data Split: We randomly select a subset of training data with $N = \{500, 2000, 5000\}$ to investigate the performance in few-sample scenario of Global Mixup.

Data	c	N	V	T
YELP	2	560000	W	38000
SST-1	5	8544	1101	2210
SST-2	2	6920	872	1821
TREC	6	5452	W	500
SUBJ	2	8500	500	1000

Table 1: Summary for the datasets c: number of target labels. N: number of samples. V: valid set size. T: test set size. W means no standard valid split was provided.

Baselines and Settings

We compare the proposed method with baselines: the original **CNNsen** (Kim 2014), the original **LSTMsen** (Hochreiter and Schmidhuber 1997), the original **BERT** (Kenton and Toutanova 2019). And two recent augmentation methods including Easy Data Augmentation(EDA) (Wei and Zou 2019), wordMixup (Guo, Mao, and Zhang 2019). CNNsen is a convolutional neural network and is widely used for text classification. LSTMsen is a type of the most popular recurrent neural network for natural language tasks. BERT is the most representative pre-training model in recent years. EDA is a simple but effective rule-based data augmentation framework for text. For a given sentence in training set, EDA (Wei and Zou 2019) randomly choose and perform one of synonym replacements, random insertion, random swap, random deletion. wordMixup (Guo, Mao, and Zhang 2019) is the straightforward application of linear interpolation on the word embedding layer. The model parameters are designed consistently to keep comparisons fair, and for comparative data augmentation methods, the best parameters from the extracted papers are used.

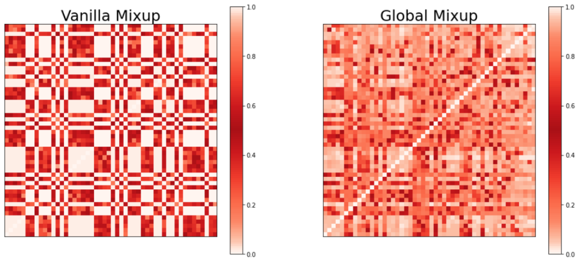

Figure 3: Heat map visualization of Global Mixup's relabeling. The coordinate axes represent the serial numbers of the original samples. The color represents the extreme degree of the label, the lighter the color, the more extreme the sample's label. The original samples are randomly selected from YELP.

Implementation Details

All models are implemented with Pytorch (Paszke et al. 2019) and Python 3.7. We set the maximum sequence length as 256 to pad the varying-length sequences. For the parameters of Global Mixup. For the $\lambda \sim Beta(\alpha, \alpha)$ parameters, we tune the α from $\{0.5, 1, 2, 4, 8\}$. And to demonstrate the effectiveness of Global Mixup on a larger space, we extend $\lambda \in [-0.3, 1.3]$ with uniform distribution. We set the number of samples generated per training sample pair T from$\{2, 4, 8, 16, 20, 32, 64\}$ and the best performance is obtained when $T = 8$ is selected. The batch size is chosen from$\{32, 50, 64, 128, 256, 500\}$and the learning rate from$\{1e-3, 1e-4, 4e-4, 2e-5\}$. For the hyperparameter setting, we set θ from$\{1/c, 0.5, 0.6, 0.8, 0.9, 1\}$, c is the number of target labels. γ from $\{1, 2, 4, 6\}$, τ and η from $\{1/T, 1\}$, $\epsilon = 1e-5$, $\delta = 1$. For the reinforced selector, we use Adam optimizer (Kingma and Ba 2015) for CNN and LSTM, AdamW (Loshchilov and Hutter 2018) for BERT. The pre-trained word embeddings for CNN and LSTM are 300-dimensional Glove (Pennington, Socher, and Manning 2014). The parameters of (Kenton and Toutanova 2019) are derived from 'bert-base-uncased'. For each dataset, we run experiments 10 times to report the mean and the standard deviation of accuracy (%).

Main experiment

To demonstrate the effect of Global Mixup, we completed the main experiment on five datasets. The main results for each dataset using CNN are shown in **Table** 2, the main results for each dataset using LSTM are shown in **Table** 3, and the main results for each dataset using BERT are shown in **Table** 4. From the result, it is clear that our method proves its effectiveness and achieves the best performance on all five datasets and three models. For instance, compared to CNNsen, Global Mixup improved the average accuracy by 3.2% on the SST-1 dataset and 2.8% on the TREC dataset. We also observe that the standard deviation of Global Mixup is smaller, which validates that Global Mixup produces more stable classification boundaries. In summary, the re-

Method	YELP	SST-1	SST-2	TREC	SUBJ
CNNsen	92.1 ± 0.44	35.3 ± 1.321	78.5 ± 0.56	95.4 ± 0.96	90.1 ± 0.43
+EDA	92.1 ± 0.24	34.1 ± 0.89	79.3 ± 0.49	97.4 ± 0.25	91.9 ± 0.21
+wordMixup	92.5 ± 0.22	36.5 ± 0.45	78.6 ± 0.36	97.8 ± 0.32	91.4 ± 0.56
+Global Mixup	**93.4 ± 0.13**	**38.5 ± 0.20**	**80.1 ± 0.23**	**98.2 ± 0.19**	**92.8 ± 0.23**

Table 2: Results(%) on five text classification tasks for **CNN**.

Method	YELP	SST-1	SST-2	TREC	SUBJ
LSTMsen	92.1 ± 0.31	36.7 ± 1.42	79.7 ± 0.64	95.2 ± 1.55	91.8 ± 0.92
+EDA	91.9 ± 0.45	37.8 ± 1.33	81.1 ± 0.66	97.6 ± 0.95	92.1 ± 0.50
+wordMixup	92.8 ± 0.22	38.4 ± 0.74	80.6 ± 0.42	98.1 ± 0.63	92.6 ± 0.42
+Global Mixup	**94.0 ± 0.16**	**39.9 ± 0.46**	**81.6 ± 0.31**	**98.6 ± 0.35**	**93.1 ± 0.39**

Table 3: Results(%) on five text classification tasks for **LSTM**.

Method	YELP	SST-1	SST-2	TREC	SUBJ
BERT	96.9 ± 0.23	51.9 ± 0.92	91.0 ± 1.16	99.0 ± 0.55	96.9 ± 0.30
+EDA	97.0 ± 0.20	51.7 ± 0.46	91.3 ± 0.55	98.5 ± 0.44	96.8 ± 0.36
+wordMixup	97.0 ± 0.13	52.0 ± 0.64	91.2 ± 0.56	99.0 ± 0.16	97.3 ± 0.32
+Global Mixup	**97.1 ± 0.15**	**52.8 ± 0.32**	**91.8 ± 0.34**	**99.2 ± 0.13**	**97.5 ± 0.35**

Table 4: Results(%) on five text classification tasks for **BERT**.

SIZE	YELP	SST-1	SST-2	TREC	SUBJ
500	74.0 ± 0.28	27.6 ± 1.11	67.8 ± 0.53	89.8 ± 1.87	83.6 ± 0.72
+Global Mixup	**81.3 ± 0.15**	**33.8 ± 0.56**	69.7 ± 0.42	**94.2 ± 0.76**	86.1 ± 0.66
2000	80.1 ± 0.32	30.8 ± 0.75	75.5 ± 0.42	93.8 ± 1.04	87.4 ± 0.45
+Global Mixup	85.6 ± 0.17	35.8 ± 0.62	77.4 ± 0.61	96.9 ± 0.51	89.4 ± 0.31
5000	85.7 ± 0.14	33.6 ± 0.78	77.6 ± 0.16	95.4 ± 0.96	88.7 ± 0.41
+Global Mixup	87.5 ± 0.13	36.6 ± 0.54	79.0 ± 0.25	98.2 ± 0.19	90.1 ± 0.32

Table 5: Results (%) across five text classification tasks with different data sizes for **CNN** with and without Global Mixup.

SIZE	YELP	SST-1	SST-2	TREC	SUBJ
500	76.0 ± 0.35	27.3 ± 1.26	68.8 ± 1.12	88.6 ± 1.65	83.4 ± 1.20
+Global Mixup	**82.1 ± 0.23**	30.0 ± 0.54	69.4 ± 0.59	90.3 ± 1.44	84.7 ± 1.03
2000	83.1 ± 0.31	35.1 ± 1.17	76.1 ± 0.98	92.5 ± 1.32	88.1 ± 1.16
+Global Mixup	87.2 ± 0.11	36.7 ± 0.56	77.8 ± 0.72	95.5 ± 0.62	89.6 ± 0.82
5000	86.2 ± 0.23	37.7 ± 1.11	79.1 ± 0.73	95.2 ± 1.55	90.2 ± 0.79
+Global Mixup	87.8 ± 0.21	38.5 ± 0.62	79.6 ± 0.43	98.6 ± 0.35	91.5 ± 0.25

Table 6: Results (%) across five text classification tasks with different data sizes for **LSTM** with and without Global Mixup.

SIZE	YELP	SST-1	SST-2	TREC	SUBJ
500	89.5 ± 0.88	35.3 ± 2.26	86.4 ± 1.42	92.6 ± 0.95	94.4 ± 0.74
+Global Mixup	91.7 ± 0.42	$\mathbf{43.0 \pm 1.15}$	88.0 ± 0.96	97.7 ± 0.66	95.0 ± 0.56
2000	92.7 ± 0.68	47.8 ± 1.55	89.2 ± 1.16	98.2 ± 0.50	95.9 ± 0.52
+Global Mixup	93.2 ± 0.42	49.0 ± 0.72	89.6 ± 0.83	98.5 ± 0.52	96.1 ± 0.36
5000	93.4 ± 0.55	51.6 ± 0.86	90.5 ± 0.63	99.0 ± 0.55	96.2 ± 0.42
+Global Mixup	94.1 ± 0.31	51.8 ± 0.55	91.2 ± 0.23	99.2 ± 0.13	96.8 ± 0.45

Table 7: Results (%) across five tasks with different data sizes for **BERT** with and without Global Mixup.

sults show a significant improvement of Global Mixup over other methods. It not only outperforms EDA, which generates real samples with rule-based, by relabeling it also outperforms wordMixup, which constructs linear relationships between samples based on linear interpolation.

In addition, as shown in the **Figure** 3, When the same sample pair is used to generate the same virtual sample, Vanilla Mixup and Global Mixup show dramatic differences, Vanilla Mixup shows very clear demarcation lines and an uneven color distribution, which indicates that it generates a large number of extreme labels for the generated samples, while Global Mixup has unclear demarcation lines and an even color distribution, which indicates that it generates almost no overconfident extreme samples. It can also be found that Global Mixup also corrected the labels of the few samples that were not overconfident, and by relabeling, the samples sometimes even obtained labels with the opposite polarity to the Vanilla Mixup labels.

Ablation Study

Effects of data size: In order to demonstrate the effect of Global Mixup in few sample scenarios, We conducted Global Mixup extended experiments with CNN, LSTM, and BERT on a few sample scenarios. The results are shown in Table 5, 6 and 7, the subset of the above data set was used for the experiments and the dataset size is set to $N = \{500, 2000, 5000\}$. Since all data of TREC dataset is 5452, the experiment of TREC with $N = 5000$ uses all data. And experiments demonstrate Global Mixup provides a greater improvement in accuracy and still effectively reduces the standard deviation. For example, when there are only 500 training samples, CNN and LSTM improved the accuracy by 7.2% and 7.3% on YELP, respectively, BERT improved the accuracy by 7.7% on SST-1. In addition, as the results in Table 5 {SST-1} show, Global Mixup exceeds the effect of training 5000 samples without data augmentation by training only 500 samples.

Effect of Different generation sample number T: We also conducted experiments on the subset of YELP to show the performance impact of the number of samples generated per original sample. As shown in **Figure** 4, Among the values $T = \{0, 2, 4, 8, 16, 20, 32, 64\}$, the range of T for which the model achieves the best results is between 4 and 20 for different sizes of datasets. In addition, relatively more improvement is achieved using Global Mixup on small datasets compared to large datasets, which we believe is due to the relatively sparse distribution of small datasets compared to large datasets, and thus the undistributed space of the training set on small datasets is relatively larger, and Global Mixup can be more useful.

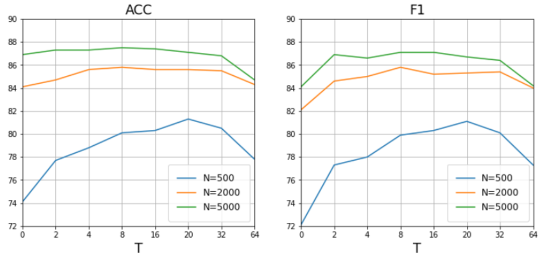

Figure 4: Ablation study on different generation sample number T. N is the size of the data set used and $\alpha = 4$. $T = 0$ means not use Global Mixup.

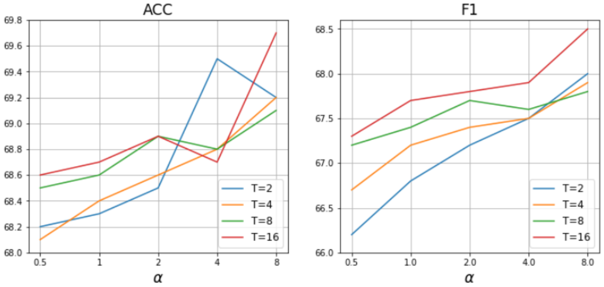

Figure 5: Ablation study on different Mixing parameter α. T is the number of samples generated from each original sample.

Effects of different Mixing parameter α: We show the performance with different α in **Figure** 5. The parameter α decides $\lambda \sim \text{Beta}(\alpha, \alpha)$, the larger α will make λ concentrate at 0.5, which means that the generated virtual samples are more likely to be further away from the original sample pairs. We choose α from $\{0.5, 1, 2, 4, 8\}$, we observed that $\alpha = 8$ achieved the best performance.

Conclusion

We propose Global Mixup, a new data augmentation method that transforms the previous one-stage augmentation process into two-stage, and solves the ambiguity problem caused by the linear interpolation of **Mixup** and **Mixup variants**. The experiment shows its superior performance, and its effect is more obvious when there are fewer samples. We believe if there is a better virtual examples generation strategy, Global Mixup will achieve better results, this is what we will explore in the future.

References

Coulombe, C. 2018. Text data augmentation made simple by leveraging nlp cloud apis. *arXiv preprint arXiv:1812.04718*.

Cui, X.; Goel, V.; and Kingsbury, B. 2015. Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(9): 1469–1477.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Guo, H.; Mao, Y.; and Zhang, R. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

Kobayashi, S. 2018. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In *NAACL-HLT (2)*.

Kober, T.; Weeds, J.; Bertolini, L.; and Weir, D. 2021. Data Augmentation for Hypernymy Detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 1034–1048.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105.

Li, X.; and Roth, D. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Loshchilov, I.; and Hutter, F. 2018. Fixing weight decay regularization in adam.

Mikolov, T.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. In *ICLR*.

Pang, B.; and Lee, L. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 271–es.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *ACL (1)*.

Simard, P.; LeCun, Y.; Denker, J. S.; and Victorri, B. 1998. Transformation Invariance in Pattern Recognition-Tangent Distance and Tangent Propagation. In *Neural Networks: Tricks of the Trade, this book is an outgrowth of a 1996 NIPS workshop*, 239–27.

Socher, R.; Bauer, J.; Manning, C. D.; and Ng, A. Y. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 455–465.

Summers, C.; and Dinneen, M. J. 2019. Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1262–1270. IEEE.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, 6438–6447. PMLR.

Wang, W. Y.; and Yang, D. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2557–2563.

Wei, J.; and Zou, K. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6382–6388.

Wu, X.; Lv, S.; Zang, L.; Han, J.; and Hu, S. 2019. Conditional bert contextual augmentation. In *International Conference on Computational Science*, 84–95. Springer.

Xie, Z.; Wang, S. I.; Li, J.; Lévy, D.; Nie, A.; Jurafsky, D.; and Ng, A. Y. 2019. Data noising as smoothing in neural network language models. In *5th International Conference on Learning Representations, ICLR 2017*.

yelp. 2015. YELP. https://www.yelp.com/dataset.

Yu, A. W.; Dohan, D.; Luong, M.-T.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. In *International Conference on Learning Representations*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhang, R.; Yu, Y.; and Zhang, C. 2020. SeqMix: Augmenting Active Sequence Labeling via Sequence Mixup. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8566–8579.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28: 649–657.