Explainable Text Classification via Attentive and Targeted Mixing Data Augmentation

Songhao Jiang[1,4,5], **Yan Chu**[2 *], **Zhengkui Wang**[3], **Tianxing Ma**[1,4], **Hanlin Wang**[2], **Wenxuan Lu**[1,4], **Tianning Zang**[1,4 *] and **Bo Wang**[5]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]Harbin Engineering University
[3]InfoComm Technology Cluster, Singapore Institute of Technology
[4]School of Cyber Security, University of Chinese Academy of Sciences
[5]CNCERT/CC
{jiangsonghao, matianxing, luwenxuan, zangtianning}@iie.ac.cn, chuyan@hrbeu.edu.cn,
zhengkui.wang@singaporetech.edu.sg, {whl10260330,wbxyz}@163.com

Abstract

Mixing data augmentation methods have been widely used in text classification recently. However, existing methods do not control the quality of augmented data and have low model explainability. To tackle these issues, this paper proposes an explainable text classification solution based on attentive and targeted mixing data augmentation, ATMIX. Instead of selecting data for augmentation without control, ATMIX focuses on the misclassified training samples as the target for augmentation to better improve the model's capability. Meanwhile, to generate meaningful augmented samples, it adopts a self-attention mechanism to understand the importance of the subsentences in a text, and cut and mix the subsentences between the misclassified and correctly classified samples wisely. Furthermore, it employs a novel dynamic augmented data selection framework based on the loss function gradient to dynamically optimize the augmented samples for model training. In the end, we develop a new model explainability evaluation method based on subsentence attention and conduct extensive evaluations over multiple real-world text datasets. The results indicate that ATMIX is more effective with higher explainability than the typical classification models, hidden-level, and input-level mixup models.

1 Introduction

Data augmentation technology can generate pseudo samples based on the given data, such as EDA [Wei and Zou, 2019], UDA [Xie *et al.*, 2020], Back-Translation [Edunov *et al.*, 2018; Sennrich *et al.*, 2016], and PromDA[Wang *et al.*, 2022]. It has been applied to many research fields. Mixing data augmentation is a powerful branch of data augmentation. It utilizes linear interpolation or clip mixing to synthesize samples and has achieved huge success in the classification tasks over continuous input characteristics such as computer vision [Zhang *et al.*, 2018; Yun *et al.*, 2019; Uddin *et al.*, 2020; Dabouei *et al.*, 2021].

Mixing data augmentation has recently drawn a lot of attention in text classification tasks as well, such as sentiment classification [Guo *et al.*, 2019; Chen *et al.*, 2020; Yoon *et al.*, 2021; Zhang *et al.*, 2022]. However, employing mixup methods in text classification remains challenging due to the discrete nature of text data and variable sequence lengths. There are two different types of mixup approaches for texts, including the hidden-level mixup and input-level mixup. The hidden-level mixup is to perform the mix operation over the hidden vectors like the word embedding, while the input-level mixup is over the input samples. Similar to computer vision, the input-level mixup becomes more promising in text classification because of its simplicity and ability to capture locality [Yoon *et al.*, 2021; Zhang *et al.*, 2022]. To augment the data, existing works select the raw data blindly or generate the pseudo samples for the mixup randomly. Such approaches result in different issues: 1) as the pseudo samples are generated without much control, a lot of useless samples may be generated for the model training. For example, if the model has already obtained a good classification capability of certain text categories, adding more mixup samples for such categories may not help (or may even harm) the model. 2) the models have low explainability. This is true as the model is trained given a large amount of "random" mixup samples without understanding which data contribute to the predictions.

To tackle the issues mentioned above, we propose an explainable text classification solution based on attentive and targeted mixing data augmentation, ATMIX. In particular, we focus on improving the training and decision-making effect of the text classification model from the perspective of model explainability. ATMIX distinguishes itself from existing approaches in multiple aspects.

First, to improve the effectiveness and explainability of the text classification models, ATMIX is empowered by a new targeted sample augmentation approach. Instead of select-

*Co-corresponding Author

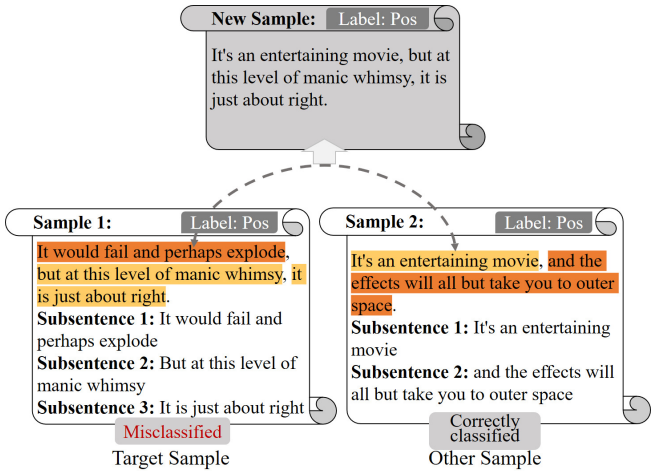

Figure 1: An example of sample generation by ATMIX for sentiment classification. The orange highlight is the attention value thermograph, and the subsentences with high attention values are darker.

ing samples for augmentation randomly, we consider taking the misclassified samples as the targets to assist the model in rectifying the decision boundary. Intuitively, the misclassified samples indicate the lack of capability for a correct prediction, where the model requires further improvement. For those correctly classified samples, the model has already achieved a good "understanding". Therefore, we try to avoid generating samples that may not contribute much to the model improvement. In fact, adding more correctly classified samples in the training may potentially introduce unnatural noise or over-fitting problems.

Second, unlike predecessors, we use a self-attention mechanism to obtain the attention distribution of the training samples' subsentences, and cut and mix the subsentences between the misclassified and correctly classified samples to generate new augmented samples wisely. We observe that many text prediction errors are caused by the neglect of the important subsentences in the training samples. As shown in Figure 1 for the sentiment classification, if the model focuses on the sample's subsentence "It would fail and perhaps explode" and ignores "It is just about right", the classification will be likely wrong. Thus, according to the attention values of subsentences, we select local text areas of the same label samples for mixed substitution to shift the model attention. The new sample generated by ATMIX is shown at the top of Figure 1.

Third, due to the difference between augmented and raw samples, some augmented samples may be ineffective and even harmful for model training. Therefore, we iteratively and dynamically choose the augmented samples through a loss gradient-based data selection algorithm in the training process, which can select the relatively optimal augmented data and help the model improve its performance. All of these approaches make the model training more effective/focused/understandable and reduce potential over-fitting problems as well.

The main contributions of this work are summarized as follows:

- We propose a targeted sample augmentation approach to focus the augmentation on the misclassified samples that the model cannot understand well.
- To generate effective augmented samples, we provide a new self-attention-based approach to understand the importance distribution of the text subsentences, and then cut and mix the subsentences based on misclassified and correctly classified samples wisely.
- We propose a loss function gradient-based dynamic augmented data selection training framework. This framework allows the model to dynamically and iteratively select the optimal augmented sample from huge augmented samples with improved performance and reduced training samples required.
- We provide a new model explainability evaluation method based on subsentence attention. We extensively evaluate our proposed model over multiple text datasets. The results indicate ATMIX outperforms typical classification models, the hidden-level and input-level mixup models w.r.t. the performance and explainability significantly.

2 Related Work

2.1 Mixing Data Augmentation

Recently, using data augmentation technology in NLP has become more prevalent [Feng *et al.*, 2021]. Some studies consider easy methods to generate new samples, such as replacing words with synonyms [Wei and Zou, 2019; Guo *et al.*, 2021], inserting punctuation marks into text [Karimi *et al.*, 2021], and so on [Wei and Zou, 2019; Guo *et al.*, 2021; Karimi *et al.*, 2021]. A powerful branch of the data augmentation methods is the mixing method which generates new samples by mixing two or more original samples. Though it has achieved huge success in computer vision (CV) [Zhang *et al.*, 2018; Guo, 2020; Yun *et al.*, 2019; Uddin *et al.*, 2020; Dabouei *et al.*, 2021], it remains challenging in NLP, due to the discrete nature of text data and variable sequence lengths. The hidden-level and input-level mixup are two different types of mixing. The hidden-level mixup attempts to mix hidden vectors like embeddings or intermediate representations to achieve text data augmentation for classification [Guo *et al.*, 2019; Chen *et al.*, 2020; Sun *et al.*, 2020]. The input-level mixup aims to mix the samples from the model input level such as SSMix [Yoon *et al.*, 2021] and TreeMix [Zhang *et al.*, 2022], which is used in ATMIX also. However, ATMIX distinguishes itself from them by using the attention value of subsentence, instead of the saliency map or text parsing tree to mix samples. Additionally, ATMIX does not need to synthesize labels, while SSMix and TreeMix have to do so. Furthermore, they have not considered model explainability and the selection of augmented data.

2.2 Selection of Augmented Data

Previous research often focuses on the scale and label of sample generation to assist in model training. However, the synthetic data may introduce harmful noise to interfere

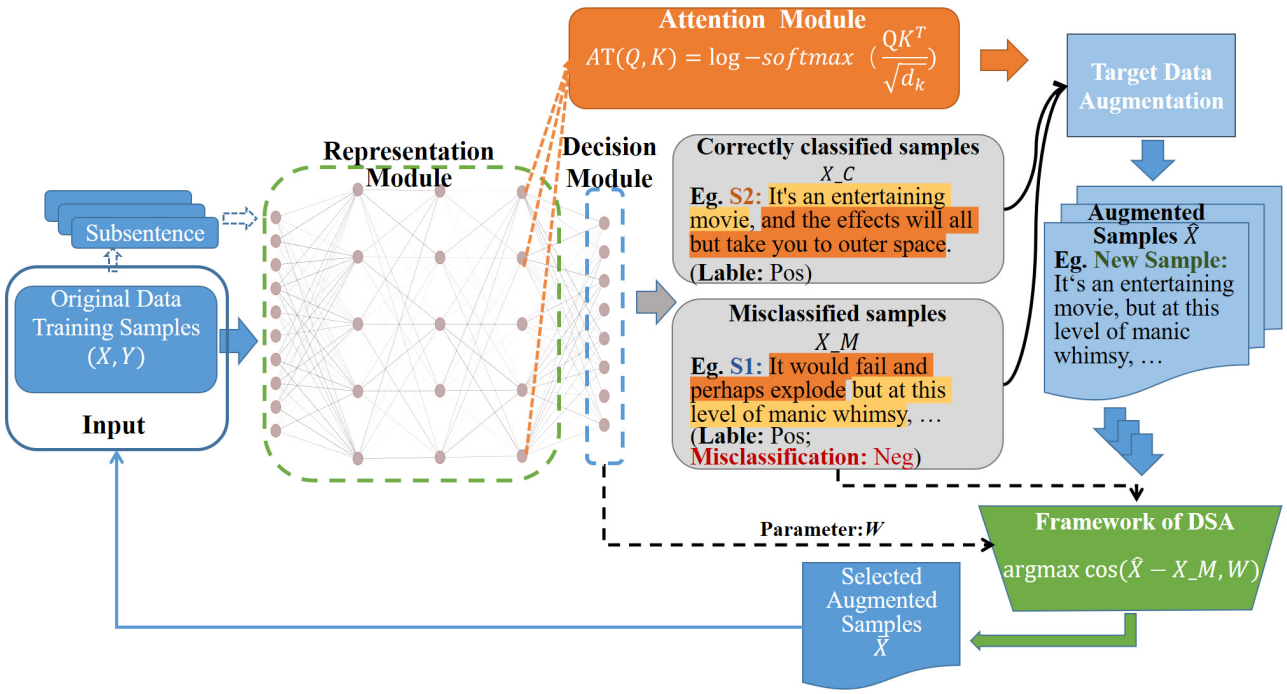

Figure 2: Illustration of the text classification model with ATMIX. The model includes six modules: (1) Input module, which is the input of original training data and selected augmented data of the model; (2) Representation module, which does not limit the specific algorithm model. Common classification models can be selected, such as BERT; (3) Decision module, which is the full connection layer and is used for the output decision of the model; (4) Attention module, which extracts the attention value of a complete sample or subsentences; (5) Data augmentation module, which generates new samples; (6) DSA module, which is used to select the optimal augmented data for training.

with model training. Some scholars try to select or generate augmented data selectively by influence function [Yang *et al.*, 2020], Monte Carlo search tree [Quteineh *et al.*, 2020], spatial distance [Sawhney *et al.*, 2022], information entropy [Zhao *et al.*, 2022] and so on [Wickramanayake *et al.*, 2021; Zhou *et al.*, 2022]. Different from the existing methods, we do not think that all raw samples can produce good results for model training by data augmentation. So we propose a concept of targeted sample augmentation to select the misclassified targeted samples to mix with correctly classified samples based on the subsentence attention, and then use the loss function to obtain the gradient value of the samples to select the augmented samples participating in the model training.

3 Methodology

We now describe our framework for attentive explanation-based text classification via targeted sample augmentation, ATMIX. Figure 2 provides the overall structure of model with ATMIX, which consists of three major components including the attention extraction of the training samples and subsentences, the data augmentation of the training samples, and the dynamic selection of augmented samples (DSA).

3.1 Subsentence Attention Acquisition

We observe that many training and classification errors are caused by the misjudgment of the important subsentences of samples in the training process. To make the model training more focused, we select the misclassified training samples as the targeted samples for augmentation in the training process. Subsequently, we cut and extract the subsentences of the training samples according to the commonly used punctuation, which may result in multiple subsentences from each training sample. And then, we use the attention module to get the attention value of subsentences. We learn from the self-attention structure of the transformer model [Vaswani *et al.*, 2017]. As shown in Figure 2, we extract the representation vector R of the sample or subsentence from the representation module. Based on training without additional parameters, we only compare the representation vector of the complete sample with the representation vector of the subsentences to obtain the attention value. The calculation equation is as follows.

$$AT(Q,K) = log\text{-}softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}), \quad (1)$$

where $AT(\cdot)$ indicates the attention function, Q is the subsentence representation vector, K^T indicates the transpose matrix of the sample representation vector K, and d_k is the dimension of the sample vector. The dot product of two vectors can be approximated as the mutual information between two vectors, which can be regarded as a co-occurrence statistic and used to calculate their semantic similarity [Levy and Goldberg, 2014; Ethayarajh *et al.*, 2019; Li *et al.*, 2020]. When the dot product between the representation vector of the complete sample and the representation

vector of the subsentences is higher, the correlation between them becomes stronger. Therefore, the higher the $AT \in [0, 1]$ value of the subsentence is, the more important the part represents.

3.2 Data Augmentation

According to the explanatory perspective of the model, it is considered that the subsentence with a higher attention value has a more important impact on the final decision-making result. To some extent, a better understanding of the importance of the subsentence results in better classification results.

To generate effective augmented data, we take the misclassified samples in the training process as the targeted samples to augment. Through data augmentation to assist the model in shifting the attention of misclassified samples to the correct position, we propose a novel method of data augmentation using text mixing. It creates a new training sample $(\hat{x}, \hat{y})$ by mixing part of the misclassified sample (x_1, y_1) and the correctly classified sample (x_2, y_2). Here, x is the training sample and y is the training label. The main idea is that the high AT subsentences of misclassified samples are replaced by the low AT subsentences of the correctly classified samples with a similar label. Defined mixing operations are as follows.

$$\hat{x} = Mix(x_1, x_2) = x_1 - A + B, \tag{2}$$

$$\hat{y} = y_1 = y_2, \tag{3}$$

$$A = \underset{sub \in sub(x_1)}{\arg\max}\ AT(R(sub(x_1)), R(x_1)), \tag{4}$$

$$B = \underset{sub \in sub(x_2)}{\arg\min}\ AT(R(sub(x_2)), R(x_2)), \tag{5}$$

where sub is a subsentence of sample x, $sub(x)$ indicates all of sample x's subsentences, and $R(x)$ indicates the representation vector of sample x. This operation is to reduce the wrong attention of subsentences of the misclassified samples. At the same time, because choosing the lowest attention subsentence of the correctly classified sample to substitute, the model would not ignore the original important subsentences again. The reason for choosing the samples with the same label for mixing is that mixing and substituting samples with the same sentiment polarity will preserve sentiment polarity most of the time [Luque, 2019]. Therefore, this work does not need to consider the label of mixed samples, and the new label $\hat{y}$ is consistent with the labels y_1 and y_2.

3.3 Dynamic Selection of Augmented Samples

Each misclassified training sample can generate a large number of new samples, according to Section 3.2. Taking the SST-1 dataset as an example, a misclassified sample will generate about 500 new samples, and the augmented scale of different datasets is different. When the scale of training data is larger, the richness of the augmented samples becomes greater. Although the scale of new samples is enormous, not every augmented sample can play a positive role in model training, because of the distribution difference between the augmented and the original sample. To this end, we propose a dynamic selection of augmented samples approach based on the gradient value of the loss function, which is applied to the augmentation samples selection of ATMIX. The augmented data selection method is provided below.

Optimal variation of samples. For the optimal selection of augmented samples, we adopt a gradient optimization algorithm to derive the optimal sample augmentation direction by the loss function $Loss(\cdot)$. We take the training data X as the variable of the loss function and the decision module parameter W as the fixed value, and optimize the derivation according to $Loss(\cdot)$, the cross entropy loss function of the model, to obtain the change gradient ∇.

$$\nabla = \frac{\partial Loss}{\partial X} = Loss'(f(R(X))) * f'(R(X)), \tag{6}$$

$$f(R(X)) = W * R(X) + b, \tag{7}$$

$$f'(R(X)) = W, \tag{8}$$

$$Loss' = (f(R(X)) - Y), \tag{9}$$

$$\nabla = W * Loss' = W * \alpha. \tag{10}$$

As shown in the equations, Y represents the label of samples, $f(\cdot)$ represents the model prediction results, and b is a constant. The gradient ∇ represents the best direction of X change of the current loss function in the case of parameter W. In the case of fixed parameter W, $f(R(X))$ and Y are fixed values, so α is a constant between -1 and 0.

Approximate optimal variation. We introduce the data gradient ascent algorithm to generate the augmented samples $\hat{X}$, which are regarded as the optimal new samples of the targeted samples X' in the gradient direction of the boundary change of the loss function, as shown in the following equation.

$$R(\hat{X}) = R(X') + \nabla, \tag{11}$$

$$\Rightarrow \nabla = R(\hat{X}) - R(X'). \tag{12}$$

Since this work adopts the mixing method in the input-level text samples, the difference in the sample vector change can not be completely consistent with the ∇. Therefore, we select the augmented sample $\hat{X}$ whose change difference of the representation vector has the closest cosine distance with the W. Since ∇ and W are approximate and in the same vector direction, $\hat{X}$ can be approximately regarded as the optimal change of X'. The DSA uses this method to carry out a selection process of the augmented samples $\hat{X}$. We define this selection operation as:

$$\bar{X} = \underset{\hat{x} \in Mix(x')}{\arg\max}\ \cos(R(\hat{X}) - R(X'), \nabla), \tag{13}$$

$$\approx \underset{\hat{x} \in Mix(x')}{\arg\max}\ \cos(R(\hat{X}) - R(X'), W), \tag{14}$$

where $\bar{X}$ is the selected augmented samples, $Mix(x')$ represents all augmented samples generated by mixing all correctly classified samples of the same category with the targeted sample $x' \in X'$.

Dynamic selection. Using the idea of the EM algorithm, the initial training of the model is carried out first. Then the data are augmented under the condition that the parameter W remains unchanged. Then the augmented samples are selected, and the chosen $\bar{X}$ is added to the next round of training data D. The DSA pseudo-code is as Algorithm 1.

Algorithm 1 Framework of DSA

Input: DATA (X, Y), Init-parameter W and θ.
Output: Classification Model M.

Training DATA $D = (X, Y)$
while epoch **do**
  $SGD(loss, M(D, \theta, W)) \Rightarrow \hat{W}, \hat{\theta}$
  $W = \hat{W}, \theta = \hat{\theta}$
  $CorrectX, ErrorX = M(X)$
  if $ErrorX! = \emptyset$ **then**
    $(\hat{X}, \hat{Y}) = \emptyset$
    for $label$ **do**
      $\hat{X}_{label} = Mix(ErrorX_{label}, CorrectX_{label})$
      $\hat{Y}_{label} = label$
      Insert $(\hat{X}_{label}, \hat{Y}_{label})$ into $(\hat{X}, \hat{Y})$
    end for
    $CosDis = \cos(R(\hat{X}) - R(ErrorX), W)$
    $(\bar{X}, \bar{Y}) = \arg\max_{(\hat{x}, \hat{y})} CosDis$
    $D = (X, Y) \cup (\bar{X}, \bar{Y})$
  end if
end while
return M

4 Experiments and Analysis

4.1 Datasets

To verify the effectiveness of ATMIX, we used five typical sentiment classification datasets, namely three two-category datasets SST-2 [Socher *et al.*, 2013], YELP-2[1], and IMDB [Maas *et al.*, 2011], and two five-category datasets SST-1 [Socher *et al.*, 2013] and YELP-5[1]. Because the amounts of YELP-2, YELP-5, and IMDB training data are relatively large, we randomly select 1% of YELP-2, YELP-5, and 20% of IMDB for the training. Whereas the amounts of the test sets are consistent with the original datasets. As shown in Table 1, we present the detailed information of five datasets, including the number of labels, the size of the training/testing set, and the average word count of each sample.

4.2 Baseline

We compare ATMIX with five baselines. (1) standard BERT without any mixing augmentation. (2) Mixup [Zhang *et al.*, 2018] applies to mix on the representation layer, which is similar to the senMixup [Guo *et al.*, 2019]. (3) TMix [Chen *et al.*, 2020] mixes hidden states of two samples at a particular layer and forwards the new states to the remaining layers. (4) SSMix [Yoon *et al.*, 2021] mixes input-level samples based on the saliency map. (5) TreeMix [Zhang *et al.*, 2022] mixes input-level samples based on the parsing tree. All mixing methods use the BERT as the backbone model. In addition, we select other two popular language models, TextCNN [Kim, 2014; Zhang and Wallace, 2015] and ALBERT [Lan *et al.*, 2019], as the backbone to verify the effectiveness of ATMIX.

[1]https://www.YELP.com/dataset

Name	Label	Train/Test Size	Word Count
SST-2	2	6920/1821	17.34
SST-1	5	8544/2210	17.19
YELP-2	2	5600/50000	135.49
YELP-5	5	6500/38000	133.41
IMDB	2	5000/25000	233.52

Table 1: The detailed statistics of the experimental datasets.

4.3 Experiment Setup

This section provides all the detailed parameter settings for different models.

For the TextCNN, ALBERT, and BERT models, we use the settings as suggested in [Kim, 2014; Zhang and Wallace, 2015; Lan *et al.*, 2019; Devlin *et al.*, 2019] to achieve the best performance. We use Tensorflow to reproduce TextCNN and use PyTorch to reproduce ALBERT and BERT. For TextCNN, we select the random initial parameter for the model and word embeddings, the dropout value is 0.5 without L2 regularization. The maximum input length is 128, and the batch size is 16. For ALBERT, we use the ALBERT-base-v2 pre-trained model from Huggingface Hub[1] for initialization. The maximum length and the batch size of the input sequence are 256 and 20 on IMDB, while 128 and 30 on the other datasets. For BERT, we use the BERT-base-uncased pre-trained model from Huggingface Hub[2] for initialization. The maximum length of the input sequence is 256 on IMDB, while 128 on the other datasets.

For other mixing methods, we try to follow the best parameter settings of the original papers for each method as well. TMix and SSMix are the repetitions of previous work [Yoon *et al.*, 2021]. We follow the best settings stated in the original papers: $\alpha = 0.2$ for Mixup and TMix, and window size is 10% for SSMix. For TMix, we randomly sample the mixing layer [7,9,12]. For TreeMix, we choose the optimal parameters of SST-2 in the original paper to test SST-1, SST-2, YELP-2 and YELP-5. The IMDB experiment for TreeMix is set to the same parameters as the original paper, except the maximum input length is 256, and no scheduler.

All experiments run on NVIDIA Tesla A100 GPUs and the epoch values are 10. The reported results are the average values for 5 runs in each experiment. For ATMIX, to speed up the efficiency of data production and selection, we randomly choose 200 correctly classified samples to mix with targeted samples. Considering the diversity of the augmented samples, we choose to replace some words in the augmented samples with synonyms. Randomly replacing a few words with synonyms can enrich the diversity of the augmented samples and retain the raw label of the training samples. Thus the augmented samples do not lead to the over-fitting problem. So for all the experiments, we use the NLTK[3] tool and the WordNet[4] corpus to replace synonyms. Each new sample replaces only 2 synonyms randomly.

[1]https://huggingface.co/ALBERT-base-v2
[2]https://huggingface.co/BERT-base-uncased
[3]https://www.nltk.org/
[4]https://wordnet.princeton.edu/

Model	SST-2	SST-1	IMDB	YELP-2	YELP-5
BERT	91.96	53.86	89.96	92.05	56.85
BERT+Mixup	92.05	53.63	89.03	92.07	56.56
BERT+TMix	92.16	53.86	90.25	92.21	56.64
BERT+SSMix	92.03	53.99	90.37	92.28	56.30
BERT+TreeMix	91.33	54.19	90.31	92.23	56.41
BERT+ATMIX($synnum = 2$)	<u>92.24</u>	**54.49**	<u>90.73</u>	<u>92.48</u>	**57.05**
BERT+ATMIX($synnum = 0$)	**92.67**	<u>54.41</u>	**90.90**	**92.61**	<u>56.96</u>
Gain	+0.62	+0.30	+0.59	+0.33	+0.20

Table 2: Performance (accuracy(%)) comparison with other text mixing methods.The best results are highlighted in bold, and the second best results are underlined. We show the gain of ATMIX for the current optimal performance.

Model	SST-2	SST-1	IMDB
TextCNN	76.63	35.08	82.83
+ATMIX(1)	**77.05**	**36.82**	82.64
+ATMIX(2)	76.36	35.73	**83.62**
ALBERT	90.12	47.49	90.41
+ATMIX(1)	91.32	**49.19**	**90.51**
+ATMIX(2)	**91.41**	48.64	90.05
BERT	91.96	53.86	89.96
+ATMIX(1)	92.07	54.43	**90.79**
+ATMIX(2)	**92.24**	**54.49**	90.73

Table 3: Performance (accuracy(%)) of the model with ATMIX($mixnum$)

Dataset	Correct	Misclassified	Full
SST-1	53.17	**54.41**	53.06
SST-2	91.96	**92.67**	91.92

Table 4: Performance (accuracy(%)) of ATMIX for different targeted samples.

4.4 Performance Comparison

We first evaluate the performance comparison between ATMIX and other text mixing methods. Table 2 shows the comparison results which provide the average results over five runs. $mixnum$ is the number of new samples selected from all augmented samples per misclassified sample. For the results shown in Table 2, we uniformly take the experimental situation when the $mixnum$ is 2. At the same time, we also show the impact of the synonym replacement quantity parameter $synnum$ on ATMIX. According to our experiment, synonym replacement has little impact on ATMIX. Even for two-category datasets such as SST-2, YELP-2, and IMDB, $synnum = 0$ performs better. In addition, we find the text mixing methods do not always show superiority over the original model for 5-category datasets, such as SST-1 and YELP-5. However, ATMIX can perform well for 5-category datasets. This result confirms that ATMIX is effective, and outperforms the original BERT model and other mixing methods.

4.5 Generalizability of ATMIX

Next, we show the proposed ATMIX can be applied to different deep learning models to further improve the model performance. Here, we choose the three most popular deep learning models (i.e., TextCNN, ALBERT, and BERT) over datasets including SST-2, SST-1 and IMDB. As a comprehensive study, we showed two different settings of using ATMIX with 1 and 2 as the $mixnum$. As shown in Table 3, with a different $mixnum$, the enhanced models perform slightly differently. However, we observe that the enhanced models with ATMIX always outperform the original models. This further confirms ATMIX can be used as a general solution over existing language models with increased performance and explainability.

4.6 Ablation Experiments

Furthermore, we provide ablation studies. We use the BERT model as the backbone model to conduct ablation experiments. We use ATMIX($mixnum = 2$,$synnum = 0$) and select correctly classified samples, misclassified samples, and full samples as targeted samples for experiments. As shown in Table 4, when the correctly classified samples and full samples are selected as targeted samples for augmentation, the performance is not better than the misclassified samples. Similarly, such results also confirm that not all the raw samples might produce a good performance for the model training by data augmentation.

To discover the influence of parameter $mixnum$ on ATMIX, we set the $mixnum$ from 1 to 5, and the experimental results are shown in Figure 3. We can find that in the two datasets, SST-1 and SST-2, with the increase of $mixnum$, the performance of ATMIX shows a decreasing fluctuation. However, the peak accuracy is achieved when $mixnum = 2$. Such results also confirm our view that adding more augmented samples to the model training might not be helpful.

In addition, to verify the effectiveness of the dynamic augmented data selection framework (DSA), we also conduct experiments on SST-1 and SST-2 and set values of $mixnum$ from 1 to 3. Instead of using DSA, we randomly select $mixnum$ samples from the generated samples. And experiments are initialized by seed 0$\sim$4 and report the average result. The results are shown in Figure 4. We can find that the effect of not using DSA is inferior to using DSA. The DSA can play a positive role in ATMIX.

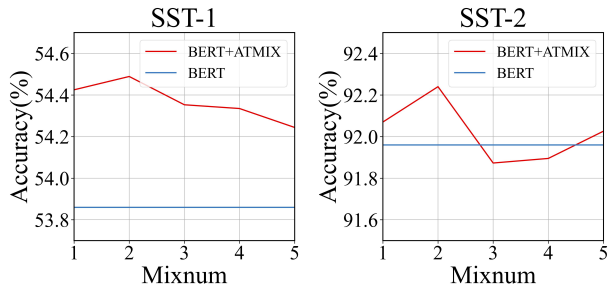

Figure 3: Influence of parameter $mixnum$ on ATMIX.

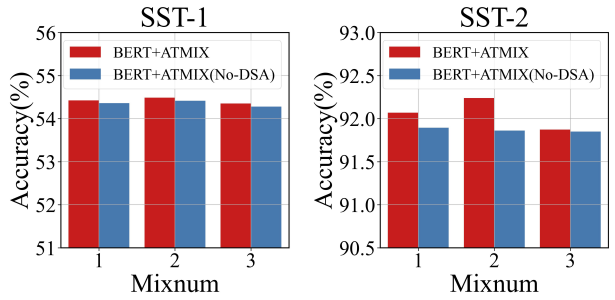

Figure 4: Influence of DSA for ATMIX.

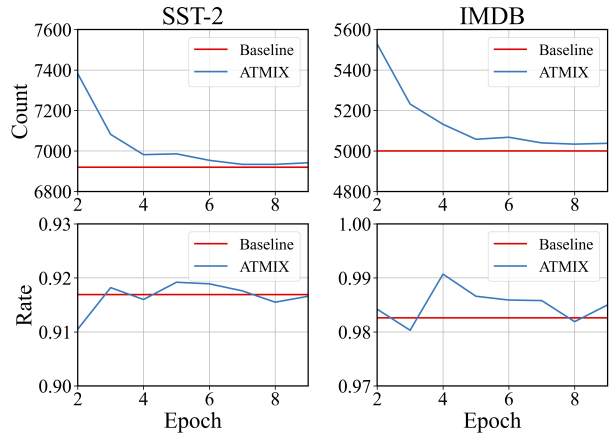

Figure 5: The above figures show the change of sample size of SST-2 and IMDB by ATMIX, and the below figures show the proportion of negative and positive samples. The baselines are the size and proportion of the original datasets.

We also track the sample count and the proportion of negative and positive samples generated in ATMIX($mixnum = 2$) training for SST-2 and IMDB. The equation for calculating the proportional value is as follows,

$$value = \frac{count(negative\ samples)}{count(positive\ samples)}. \tag{15}$$

As shown in Figure 5, the lines represent the change of values of Baseline and ATMIX with the training epoch. The samples generated by ATMIX gradually decrease with the increase of epoch and have limited influence on the distribution of negative and positive samples. The proportion of deviation from the original distribution of negative and positive samples could not exceed 1% at most. Therefore, we believe that ATMIX can not cause sample imbalance.

4.7 Model Explainability

Explainability becomes important to measure how the model can be interpreted and understood. There are multiple interpretable models proposed, such as SHAP [Lundberg and Lee, 2017], LIME [Ribeiro *et al.*, 2016], Anchor [Ribeiro *et al.*, 2018] and word cosine distance [Chen and Ji, 2019]. Differently, ATMIX provides a new capability to evaluate the model explainability from the perspective of self-attention by further understanding the subsentences in a text towards the classification. Figure 2 illustrates the overall architecture of ATMIX, where the attention module forms a self-attention mechanism independent of model parameters or weights. During training, the self-attention mechanism dynamically monitors the attention distribution of training samples and their subsentences, particularly for misclassified samples. Based on the attention value obtained, the high attention subsentence of the misclassified sample is replaced with a low attention subsentence of correctly classified samples to generate an augmented sample for model training, and to correct the model's attention distribution for misclassified samples. During the testing phase, the attention module of ATMIX can visualize the model's attention distribution of the test sample subsentences and obtain insights into its functioning.

While the existing interpretable models cannot explain the model directly by capturing the attention of subsentences. To this end, we present a new method to evaluate the model's explainability. By comparing the top important subsentences for the model with the human-labeled subsentences, we calculate the accuracy of the model's understanding of the subsentences. The indicators equation is as follows.

$$Att_{top\ n} = \arg\max_{top\ n}(AT(subs)), \tag{16}$$

$$Ann_{top\ m} = Annotated_{top\ m}(subs)), \tag{17}$$

$$ACC@n@m = \frac{Count(Att_{top\ n} \cap Ann_{top\ m})}{min(n, m) * 200}. \tag{18}$$

where $AT(subs)$ is shown in Equation 1 and indicates the model attention values of subsentences, $Annotated(subs)$ indicates the result of manual annotation, n represents the number of the top essential subsentences in each sample judged by the model, and m represents the number of the top important subsentences in each sample labeled manually.

We manually label 200 samples in the test sets of SST-2 and IMDB respectively. There are many subsentences in each sample in IMDB. Therefore, we label three IMDB training subsentences that affect the sentiment polarity according to our understanding of the importance degree. However, there are fewer subsentences in SST-2, and we only label the subsentence that affects the sample sentiment polarity mostly.

We use BERT as the backbone model to evaluate the subsentence attention of the original model and the four text mixing augmentation methods (Mixup, TMix, SSMix, and ATMIX). As shown in Table 5, for SST-2, because each sample has an average of 2 subsentences, we use ACC@1@1 as

	SST-2	IMDB	
Model	@1@1	@1@1	@3@3
BERT	69.50	21.90	48.00
+Mixup	68.10	22.00	45.90
+TMix	63.60	18.10	42.40
+SSMix	65.10	20.20	42.90
+ATMIX	**70.00**	**22.30**	**48.70**

Table 5: The ACC@n@m(%) of the model capture the important subsentence. The best results are highlighted in bold.

an indicator to evaluate the accuracy of subsentences' importance understood by the model. The results show ATMIX can improve the subsentence importance understanding significantly. For IMDB, as samples are longer and have more subsentences, we use ACC@1@1 and ACC@3@3 as the indicators. We observe ATMIX is also the best in extracting important subsentences. These results further confirm ATMIX can pay more attention to the most important subsentences.

5 Conclusion

Mixing data augmentation has emerged as a promising solution for text classification. In this paper, we advanced the concept of targeted sample augmentation. And we proposed an explainable text classification solution based on attentive and targeted mixing data augmentation, ATMIX. In particular, we focused on the misclassified samples as the candidates to generate new augmented data to better improve the model's capability in understanding these data. To generate the most meaningful augmented data, we also provided a self-attention-based mechanism to capture the importance of different subsentences in the model and used a cut-and-mix approach to mix the subsentences between the correctly classified and misclassified samples. Among a large amount of augmented data samples, we further proposed a loss function gradient-based dynamic data selection training framework, to dynamically select the optimal augmented samples to improve model performance with reduced training samples needed. In the end, we also provided a new model explainability evaluation method based on subsentence attention. To evaluate the performance and explainability of the solution, we performed extensive evaluations with current popular classification models, the models with hidden-level and input-level mixup models. Our results confirmed that the proposed solution outperforms these models significantly in terms of performance and explainability.

In the future, We will continue to study the selection of target data and consider upgrading the framework of DSA. We will consider applying DSA to other data augmentation methods for experiments and expanding it to other research fields. In addition, we will study optimizing the overall calculation process of DSA and searching for the best enhancement selection parameters automatically.

Acknowledgements

This work is supported by the Fundamental Research Funds for the Central Universities Grant (3072022TS0601), the National Natural Science Foundation of China Grant (61771155), the Singapore Institute of Technology Ignition Grant (R-IE2-A405-0001), the Heilongjiang Postdoctoral Foundation Grant (LBH-Z22103), and the China National Key Laboratory Foundation of Underwater Measurement and Control Technology.

References

[Chen and Ji, 2019] Hanjie Chen and Yangfeng Ji. Improving the explainability of neural sentiment classifiers via data augmentation. *arXiv preprint arXiv:1909.04225*, 2019.

[Chen *et al.*, 2020] Jiaao Chen, Zichao Yang, and Diyi Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, 2020.

[Dabouei *et al.*, 2021] Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, and Nasser M Nasrabadi. Supermix: Supervising the mixing data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13794–13803, 2021.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[Edunov *et al.*, 2018] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, 2018.

[Ethayarajh *et al.*, 2019] Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Towards understanding linear word analogies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3253–3262, 2019.

[Feng *et al.*, 2021] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, 2021.

[Guo *et al.*, 2019] Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*, 2019.

[Guo *et al.*, 2021] Biyang Guo, Sonqiao Han, and Hailiang Huang. What have been learned & what should be learned? an empirical study of how to selectively augment text for classification. *arXiv preprint arXiv:2109.00175*, 2021.

[Guo, 2020] Hongyu Guo. Nonlinear mixup: Out-of-manifold data augmentation for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4044–4051, 2020.

[Karimi *et al.*, 2021] Akbar Karimi, Leonardo Rossi, and Andrea Prati. Aeda: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, 2021.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.

[Lan *et al.*, 2019] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[Levy and Goldberg, 2014] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27, 2014.

[Li *et al.*, 2020] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*, 2020.

[Lundberg and Lee, 2017] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[Luque, 2019] Franco M Luque. Atalaya at tass 2019: Data augmentation and robust embeddings for sentiment analysis. *arXiv preprint arXiv:1909.11241*, 2019.

[Maas *et al.*, 2011] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

[Quteineh *et al.*, 2020] Husam Quteineh, Spyridon Samothrakis, and Richard Sutcliffe. Textual data augmentation for efficient active learning on tiny datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7400–7410. Association for Computational Linguistics, 2020.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[Ribeiro *et al.*, 2018] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[Sawhney *et al.*, 2022] Ramit Sawhney, Megh Thakkar, Shrey Pandit, Ritesh Soun, Di Jin, Diyi Yang, and Lucie Flek. Dmix: Adaptive distance-aware interpolative mixup. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 606–612, 2022.

[Sennrich *et al.*, 2016] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, 2016.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[Sun *et al.*, 2020] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, S Yu Philip, and Lifang He. Mixup-transformer: Dynamic data augmentation for nlp tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3436–3440, 2020.

[Uddin *et al.*, 2020] AFM Shahab Uddin, Mst Sirazam Monira, Wheemyung Shin, TaeChoong Chung, and Sung-Ho Bae. Saliencymix: A saliency guided data augmentation strategy for better regularization. In *International Conference on Learning Representations*, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2022] Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. Promda: Prompt-based data augmentation for low-resource nlu tasks. *arXiv preprint arXiv:2202.12499*, 2022.

[Wei and Zou, 2019] Jason Wei and Kai Zou. 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)*, pages 6382–6388, 2019.

[Wickramanayake *et al.*, 2021] Sandareka Wickramanayake, Wynne Hsu, and Mong-Li Lee. Explanation-based data augmentation for image classification. In *Advances in Neural Information Processing Systems*, 2021.

[Xie *et al.*, 2020] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

[Yang *et al.*, 2020] Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug

Downey. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, 2020.

[Yoon *et al.*, 2021] Soyoung Yoon, Gyuwan Kim, and Kyumin Park. Ssmix: Saliency-based span mixup for text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3225–3234, 2021.

[Yun *et al.*, 2019] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[Zhang and Wallace, 2015] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

[Zhang *et al.*, 2018] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[Zhang *et al.*, 2022] Le Zhang, Zichao Yang, and Diyi Yang. Treemix: Compositional constituency-based data augmentation for natural language understanding. *arXiv preprint arXiv:2205.06153*, 2022.

[Zhao *et al.*, 2022] Minyi Zhao, Lu Zhang, Yi Xu, Jiandong Ding, Jihong Guan, and Shuigeng Zhou. Epida: An easy plug-in data augmentation framework for high performance text classification. *arXiv preprint arXiv:2204.11205*, 2022.

[Zhou *et al.*, 2022] Jing Zhou, Yanan Zheng, Jie Tang, Li Jian, and Zhilin Yang. Flipda: Effective and robust data augmentation for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8646–8665, 2022.