ASMix: An Attention-based Smooth Data Augmentation Approach

Rongxi Guo, Wanrong Jiang, Xi Zhao, Ding Cao, Guiquan Liu*
University of Science and Technology of China, Hefei, China
{guorongxi, jwr, xixilili, caoding}@mail.ustc.edu.cn, gqliu@ustc.edu.cn

***Abstract*—Data augmentation through linearly interpolating inputs and modeling targets of random samples has significantly improved predictive performance. However, data augmentation based on linear interpolation generates semantically cluttered and ambiguous text, resulting in ineffective augmentation. To address these issues, in this paper, we propose a novel data augmentation approach called Attention-based Smooth Data Augmentation (ASMix). ASMix accepts the smoothed embeddings of pairwise data predicted by a masked language model (MLM) instead of one-hot embeddings, which makes the inputs more informative and context-rich. We employ the attention mechanism to select discriminative and more-attentioned parts of text hidden representations and mix up the parts containing key semantics in the hidden representations of pairwise data through a multi-token replacement strategy to augment the data of the minority class, which greatly reduces redundant information in the representations that hurts the performance of the model. On several public imbalanced text classification benchmarks, ASMix outperforms state-of-the-art data augmentation methods. In minority classes, the performance improvement of ASMix is particularly prominent.[1]**

***Keywords*—Data imbalance, Data augmentation, Text classification**

I. Introduction

One of the challenges of machine learning is the lack of sufficient data, which leads to data imbalance problems. In the imbalanced binary classification task, the class with the larger size is known as the majority class, and the other class is known as the minority class [1].

In the case of imbalanced data, traditional classifiers learn biased models that tend to be biased towards the majority class and overfit the minority class. Many methods have been proposed. Resampling [2] has received extensive attention for its simplicity and effectiveness, including undersampling and oversampling. As an improved oversampling method, data augmentation addresses the lack of data by increasing the amount of training data. Mixup [3] mixes two samples by interpolating their images and labels to generate a virtual sample as training data in Computer Vision (CV). CutMix [5] replaces image regions with a patch of another training image to overcome local ambiguity and unnaturalness caused by interpolation.

Due to the discreteness of the text input space and the complexity of the text structure, it is challenging for CutMix to apply in the text input space. If we apply it to the hidden layer, there is still a problem. Many tokens in the text are irrelevant to classification, such as stop words with no actual meaning, meaningless filler words in padding to fill the sentence length to the maximum length, and words containing other information unrelated to classification. These tokens are unimportant for text classification tasks and may also hurt the performance of classifiers. If we randomly replace the hidden representations of tokens, these stop words, filler words, and tokens unrelated to classification may be incorporated into synthetic samples, making samples incredible.

To overcome the above shortcomings, in this work, we propose a novel augmentation method called ASMix. ASMix accepts the smoothed embeddings of pair-wise data predicted by MLM instead of one-hot embeddings as model inputs, which are more informative and context-rich. [6] In the hidden space, to avoid meaningless or irrelevant information, ASMix uses the self-attention mechanism [7] to select the text information that the model pays more attention to through a multi-token replacement strategy. ASMix fully mixes the semantics related to classification in the pairwise data and generates new samples that are not similar to the original samples, which makes the decision boundary smoother and further away from the training data.

Our main contributions in this article are as follows:

- We propose a novel data augmentation method that employs a self-attention mechanism to combine pairwise smoothed samples through a local replacement strategy to generate augmented samples.
- To the best of our knowledge, we are the first to mix label-related tokens of sentences in the hidden layer to enhance the text.
- Our proposed ASMix achieves state-of-the-art performance on three imbalanced classification datasets, is particularly helpful for the minority class, and shows robustness across languages.

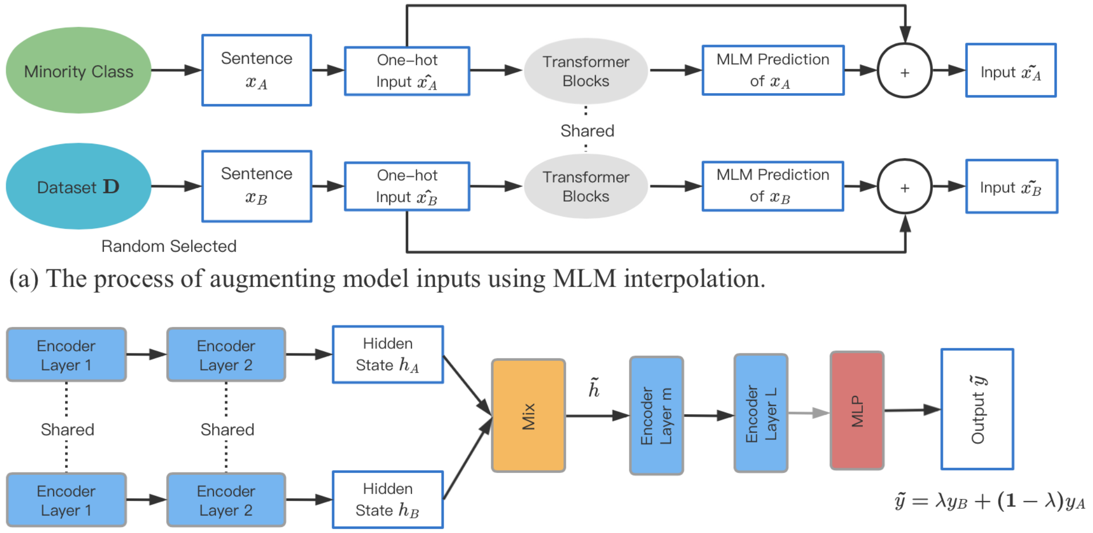
II. Related Work

A. Imbalanced Learning

The current methods of imbalanced learning are divided into two categories: algorithm-level methods and data-level methods. The cost-sensitive method [8] is an algorithm-level approach that takes the costs into account, which improves the classifiers by assigning different costs to classes. In addition, Focal Loss [9] alleviates the class imbalance problem by

[1] Our code is available at: https://github.com/GuoRongxi/ASMix

DOI reference number: 10.18293/SEKE2023-191



(a) The process of augmenting model inputs using MLM interpolation.

(b) The overall framework for imbalanced text classification using ASMix.

Fig. 1: The overall architecture of ASMix.

modifying the Cross-entropy loss function to assign lower weights to the loss of easily classified samples. Resampling [2] is a data-level method to deal with the problem of data imbalance. SMOTE [10] is a classic oversampling method that takes the linear interpolation of minority class samples and their nearest neighbors as synthetic samples.

B. Data Augmentation for Text

Data augmentation was initially widely used in CV and has since been extended to Natural Language Processing (NLP) by many researchers. Back-translation [11] generates different data on the premise of keeping the semantics of the original sentence unchanged. Guo et al. [12] first introduced Mixup [3] to the NLP task. TMix [4] synthesizes a large amount of augmented training data by interpolating in Euclidean space, while HYPMIX [13] performs interpolation operations in hyperbolic space to better capture the complex geometry of hidden state hierarchies. Previous data augmentation methods expanded all categories. Our proposed ASMix not only expands all data but also mainly augments the minority class, which can better address the data imbalance problem.

III. Proposed Method

A. Notations

Given a text dataset, $\mathbf{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where N is the number of instances, x_i is the discrete sequence of text in the input space, y_i is the label of the instance, and the number of classes in dataset $\mathbf{D}$ is n. (x_A, x_B, y_A, y_B) represents a pair of data extracted from the dataset $\mathbf{D}$, where $A \neq B$.

B. Representation Augmentation

BERT [14] alleviates the unidirectionality constraint by proposing a “masked language model” pre-training objective. The masked language model masks some percentage of the input tokens at random, and the objective is to predict the original vocabulary of the masked word based only on its context.

Given an instance (x_A, y_A), we first convert the original text x_A into an one-hot encoding $\hat{x_A}$ by tokenizer. We use the one-hot encoding $\hat{x_A}$ as the input of BERT and obtain the output of the last layer of the transformer encoder in BERT, which is denoted as:

$$\hat{x_A} = \text{convert\_to\_onehot}(\text{tokenizer}(x_A)) \quad (1)$$

$$\vec{x_A} = \text{BERT}(\hat{x_A}) \quad (2)$$

where $\vec{x_A} \in \mathbb{R}^{\text{seq\_len} \times \text{emb\_size}}$ is a 2D dense vector. We multiply $\vec{x_A}$ and the word embedding matrix $W \in \mathbb{R}^{\text{vocab\_size} \times \text{emb\_size}}$ in BERT to get the prediction result of MLM, which is mainly distributed over the context-compatible tokens at this position. To address the model's preference for tokens that appear in similar contexts but conflict with task labels, we employ the interpolation, which is defined as:

$$\text{MLM}(x_A) = \text{softmax}(\vec{x_A} W^T) \quad (3)$$

$$\tilde{x_A} = \mu \hat{x_A} + (1 - \mu)\text{MLM}(x_A) \quad (4)$$

where $\tilde{x_A}$ is the interpolated representation, and μ is the balance hyperparameter that controls the interpolation strength. We use $\tilde{x_A}$ as the input to the classifier instead of the one-hot representation. The one-hot representation $\hat{x_A}$ and smoothed representation $\text{MLM}(x_A)$ are derived from the same raw input x_A. We keep the label y_A unchanged.

C. ASMix

Given a pair of enhanced representation inputs $(\tilde{x_A}, \tilde{x_B}, y_A, y_B)$, we compute the hidden representations $h_A \in \mathbb{R}^{l \times d}$ and $h_B \in \mathbb{R}^{l \times d}$ of the inputs $\tilde{x_A}$ and $\tilde{x_B}$ separately in the bottom layers of the model, where l is the

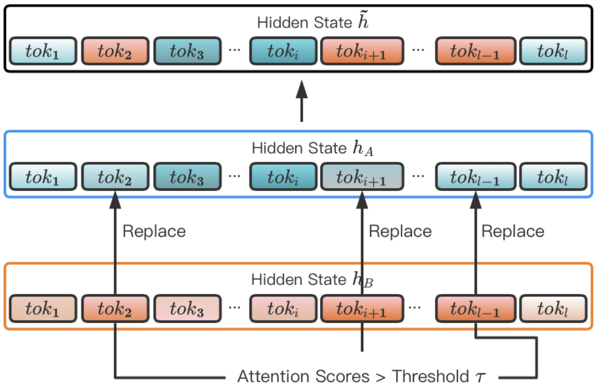


Fig. 2: Illustration of the multi-token replacement strategy.

maximum length of the sentence and d is the dimension of the hidden representation of each token.

Similar to CutMix, we define the combining operation as:

$$\begin{aligned}\tilde{h} &= (\mathbf{1} - \mathbf{M}) \odot h_A + \mathbf{M} \odot h_B \\ \tilde{y} &= (1 - \lambda) y_A + \lambda y_B\end{aligned} \tag{5}$$

where $\mathbf{M} \in \{0, 1\}^{l \times d}$ denotes a binary mask indicating where to drop out and fill in from two samples introduced in the following section, λ represents the interpolation strength of the one-hot representation of the labels, and $\odot$ is element-wise multiplication. The interpolation strength λ is calculated using the number of tokens participating in ASMix, which is consistent with the combination ratio of the representations.

D. Multi-token Replacement Strategy

We use the scaled dot-product attention [7] to explore the dependency between the target and candidates from the hidden state h and compute the total attention score for each token in a sequence as follows:

$$\mathbf{A} = h(\mathrm{softmax}((\mathrm{W_q} h)^T (\mathrm{W_k} h))) \tag{6}$$

$$\mathbf{A}_i^* = \sum_{j=1}^{d} \mathbf{A}_{ij} \tag{7}$$

where $\mathrm{W_q}$ and $\mathrm{W_k}$ are trainable weights, $\mathbf{A} \in \mathbb{R}^{l \times d}$ is the output of the attention layer, and $\mathbf{A}_i^*$ represents the attention score of the i-th token, $i \in [1, l]$. We sample the binary mask $\mathbf{M}$ as follows:

$$\begin{aligned}\mathbf{M}_i &= \begin{cases} \mathbf{1}^{1 \times d} & if\ \mathbf{A}_i^* \geq \tau \\ \mathbf{0}^{1 \times d} & otherwise \end{cases} \\ \mathbf{M} &= \mathrm{Concat}(\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_l)\end{aligned} \tag{8}$$

where $\mathrm{Concat}(.)$ is the concatenate operation, τ is the attention score threshold. We employ a random sampling strategy to obtain the threshold τ as follows:

$$\begin{aligned}\tau &= \theta \max_i (\mathbf{A}_i^*) \\ \theta &\sim \mathrm{Beta}(\alpha, \alpha)\end{aligned} \tag{9}$$

Here, the parameter θ is a random value sampled from a Beta distribution, and α is a hyper-parameter that controls the distribution of θ. The threshold τ determines which tokens in the hidden representation will be combined in ASMix.

TABLE I: Dataset statistics and dataset split.

Dataset	Class	Train	Test	IR
R8	earn	2673	1040	-
	acq	1438	637	1.859
	trade	230	64	11.622
	crude	222	113	12.041
	money-fx	168	69	15.911
	interest	144	66	18.563
	ship	107	35	24.981
	grain	37	10	72.243
Cade12	servicos	5627	2846	-
	sociedade	4935	2428	1.140
	lazer	3698	1892	1.522
	internet	1585	796	3.550
	noticias	701	381	8.027
	compras-online	423	202	13.303
THS	0	20793	8914	-
	1	1571	670	13.236

In this method, we select several tokens with discriminative information in the hidden states of the instance x_B randomly sampled from the training set to replace parts of h_A to augment (x_A, y_A), and the interpolation strength λ tends to be less than 0.5. In experiments, we resample θ to ensure λ is less than 0.5. In ASMix, the deleted tokens may also attract attention in h_A. However, these tokens occupy only a small portion, and the remainder still contains the majority of vital information in h_A. Therefore, we ignore this problem in this work.

E. Framework of Text Classification

As shown in Fig. 1, we implement ASMix at one of the layers of BERT. For an imbalanced text dataset $\mathbf{D}$, n classes are represented as $\{C^0, ..., C^{n-1}\}$, where C^0 is the majority class with the most samples. For the other class C^i, $i \in [1, n-1]$, the Imbalance Ratio (IR) of the class is the ratio of $size(C^0)$ and $size(C^i)$.

For a class C^i, $i \in [1, n-1]$, we determine the sampling ratio $r = \lfloor \mathrm{IR} - 1 \rfloor$. For a sample x in the class C^i, we randomly sample r samples $\{x_1^{aug}, ..., x_r^{aug}\}$ in the training set $\mathbf{D}$, resulting in r sample pairs $\{(x, x_1^{aug}), ..., (x, x_r^{aug})\}$. For the majority class C^0, there is no need to sample new samples, $x^{aug} = x$.

We use the interpolation representation $(\tilde{x}, \tilde{x}^{aug})$ of a text pair (x, x^{aug}) as the input of BERT, which is transformed into a pair of hidden representations (h, h^{aug}). As an encoder, BERT has L layers. We choose to employ ASMix at the m-th layer, where $m \in [0, L]$. The l-th layer in the network is represented as $f_l(.;\theta)$. The hidden representation of layer l can be computed with $h_l = f_l(h_{l-1};\theta)$. First, we compute the hidden representations of the two text samples separately in the bottom layers of the model:

$$\begin{aligned}h_l &= f_l(h_{l-1};\theta), l \in [1, m] \\ h_l^{aug} &= f_l(h_{l-1}^{aug};\theta), l \in [1, m]\end{aligned} \tag{10}$$

Then, we execute ASMix at the m-th layer and continue to pass the augmented representation to the upper layers of

TABLE II: Performance (precision (%), recall (%), F1 (%)) of all categories in comparison with baselines.

Model	R8			Cade12			THS		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
BERT	95.22	92.97	94.01	62.69	56.74	58.75	91.54	87.26	89.26
EDA	95.46	94.20	94.83	60.43	58.85	59.63	93.56	90.21	91.86
AEDA	95.71	94.03	94.86	61.77	57.93	59.78	93.13	89.74	91.40
TMix	95.23	94.76	94.99	60.44	57.83	59.10	94.23	90.65	92.40
HYPMIX	95.62	94.22	94.92	59.53	56.45	57.95	94.44	90.78	92.57
ASMix	**95.82**	**95.58**	**95.70**	**63.36**	**58.93**	**61.07**	**95.25**	**91.21**	**93.19**

the model, the $(m+1)$-th to L-th layers. The labels are also interpolated with the strength consistent with the combination ratio of the hidden states. The definition is as follows:

$$\begin{aligned} \tilde{h}_m &= (\mathbf{1}-\mathbf{M}) \odot h_m + \mathbf{M} \odot h_m^{aug} \\ \tilde{h}_l &= f_l(\tilde{h}_{l-1};\theta), l \in [m+1, L] \end{aligned} \tag{11}$$

In text classification, we implement the classifier as a two-layer MLP following BERT. It takes the representation of the BERT output as input and returns a probability vector. We train the entire model by minimizing the Cross-entropy between the interpolated labels and the probability from the classifier as follows:

$$\begin{aligned} L = -\sum_{\gamma} (\tilde{y}_\gamma \log \mathbf{ASMix}(\tilde{x}_\gamma, \tilde{x}_\gamma^{aug}) \\ + (1-\tilde{y}_\gamma)\log(1-\mathbf{ASMix}(\tilde{x}_\gamma, \tilde{x}_\gamma^{aug}))) \end{aligned} \tag{12}$$

where $\mathbf{ASMix}(\cdot,\cdot)$ represents the output of the classifier.

IV. Experiments

A. Datasets

We conducted experiments on three benchmark imbalanced text datasets.

- **R8** [15] is a corpus of Reuters news articles in English containing 29,930 words.
- **Cade12** [15] is a corpus of Brazilian web pages classified by human experts extracted from the CADÊ Web Directory.
- **THS** [16] is an imbalanced binary dataset for Twitter hate speech detection. The label '1' denotes the tweet is racist or sexist, and the label '0' represents the tweet is neither racist nor sexist.

We used the original training and testing sets for the first two datasets as our training and testing sets. For the last dataset, we split the dataset into training and test sets at a ratio of 7:3. The dataset statistics and split information are presented in Table I.

B. Baselines

To test the effectiveness of our method, we compared it with several recent methods. The following methods all used BERT multilingual base model as the backbone network:

- **BERT** [14]: We used a pre-trained BERT-base-multilingual-cased model and fine-tuned it for classification.
- **EDA** [17]: Easy Data Augmentation (EDA) chooses and performs one of the following operations at random for each sentence in the training set: Synonym Replacement (SR), Random Insertion (RI), Random Swap (RS), and Random Deletion (RD).
- **AEDA** [18]: An Easier Data Augmentation (AEDA) randomly inserts punctuation marks into the original text.
- **TMix** [4]: TMix takes two text examples and interpolates them in their corresponding hidden space.
- **HYPMIX** [13]: HYPMIX is an interpolative data augmentation technique operating in the hyperbolic space.

C. Experimental Settings

We used BERT-base-multilingual-cased tokenizer to tokenize the text, BERT-base-multilingual-cased model as our text encoder, average pooling on the output of the encoder, and a two-layer MLP with a 128 hidden size and $tanh$ as its activation function. We used AdamW as the optimizer and the weight decay is 0.01. The learning rates for BERT encoder and MLP are set to 1e-5 and 1e-3, respectively. We train all models for 50 epochs and set the batch size to 16. The number of heads of the multi-head attention mechanism, t, is set to 8. For R8, Cade12, and THS, the maximum sentence length is 128, 150, and 50, respectively. In addition, we set a fixed seed when training the model to ensure the reproducibility of the results.

D. Overall Results

We used Precision (%), Recall (%), and macro-F1 (%) as metrics to evaluate our method ASMix by comparing with all the baselines mentioned above on different imbalanced text classification datasets.

The overall results on different imbalanced text classification datasets are shown in Table II. Firstly, basically all models using data augmentation outperform BERT on the F1 metric. Secondly, our method outperforms EDA, AEDA, TMix, HYPMIX in all metrics. Compared with other interpolation-based methods, TMix and HYPMIX, ASMix avoids meaningless information, resulting in a sample that more thoroughly combines the semantics of the two original instances.

E. Analysis of Each Category

To show the effect of our method on enhancing the minority class in the imbalanced text classification dataset, Table ta-

TABLE III: Performance (F1 (%)) of each category on R8 in comparison with baselines.

Model	earn	acq	trade	crude	money-fx	interest	ship	grain
BERT	98.25	97.29	89.76	92.45	89.36	92.42	92.54	100
EDA	97.96	97.94	88.36	92.97	91.17	**95.28**	93.94	100
AEDA	98.70	**98.12**	**92.91**	91.51	88.89	93.85	94.12	100
TMix	**98.81**	98.02	92.42	93.09	92.09	91.04	94.12	100
HYPMIX	98.76	97.72	91.04	91.00	**92.75**	91.73	95.52	100
ASMix	98.52	97.15	90.08	**94.51**	92.70	93.91	**98.55**	100

TABLE IV: Performance (F1 (%)) of each category on Cade12 in comparison with baselines.

Model	servicos	sociedade	lazer	internet	noticias	compras-online
BERT	73.14	**67.23**	66.10	56.56	43.67	45.76
EDA	68.89	66.58	66.56	54.51	**52.65**	47.68
AEDA	72.00	64.36	**67.47**	59.22	48.07	45.86
TMix	69.72	64.10	65.52	62.13	45.58	47.11
HYPMIX	69.25	63.10	64.36	60.48	48.10	41.28
ASMix	**74.94**	60.41	64.12	**62.21**	50.56	**52.83**

TABLE V: Performance (F1 (%)) of each category on THS in comparison with baselines.

Model	0	1
BERT	**98.58**	79.94
EDA	96.33	87.37
AEDA	97.79	84.92
TMix	94.57	90.22
HYPMIX	96.37	88.61
ASMix	95.54	**90.84**

TABLE VI: Performance(F1(%)) on three datasets after removing different parts of ASMix.

Model	**F1(%)**		
	R8	Cade12	THS
ASMix	**95.70**	**61.07**	**93.19**
- smoothed	95.36	60.75	93.07
- threshold	94.51	59.80	92.86
- attention	94.87	59.78	91.96
- all	94.01	58.75	89.26

bles III to V show the F1 of each class on R8, Cade12, and THS, respectively.

The IR of each class on these datasets is shown in Table I. Firstly, BERT shows that the classification performance gradually decreases as the IR of the class keeps increasing because the model will be biased towards the majority class. Secondly, ASMix performed best in severe imbalanced categories in all three datasets. ASMix has a slight drop in categories with smaller IR on the F1 metric. It is natural for ASMix to trade off when ASMix focuses more on the minority class.

F. Varying the Number of Data

We evaluated our baselines and proposed methods using F1 with different fractions of the available training data ranging from 10 to 100 percent. The results on different text classification datasets are shown in Fig. 3. We show that ASMix consistently demonstrated the best performances when compared to different baseline models across three datasets with different fractions of the data. On Cade12, the best F1 without augmentation, 58.7%, was achieved using 100% of the training data. ASMix surpassed this number by achieving an F1 of 59% while only using 70% of the available training data.

G. Ablation Studies

We performed ablation studies to show the effectiveness of each component in ASMix. We measured the performance of ASMix by stripping each component each time and displayed the results in Table VI. We observed a drop in performance after removing each part, suggesting that all components in ASMix contribute to the final performance. Overall, the model performance dropped most significantly after removing the self-attention mechanism, which verified the effectiveness of the multi-token replacement strategy.

H. Parameter Studies

All parameter experiments use automatic mixed precision(AMP) of Pytorch, which saves memory and speeds up.

1) Hyper-parameter of the Beta Distribution α: We varied the hyper-parameter α in $\{0.25, 0.5, 0.75, 1\}$. Fig. 4 shows ASMix achieved optimal values on all three datasets when α was small. From the perspective of Beta distribution, smaller α leads to a lower threshold τ and λ closer to 0.5, resulting

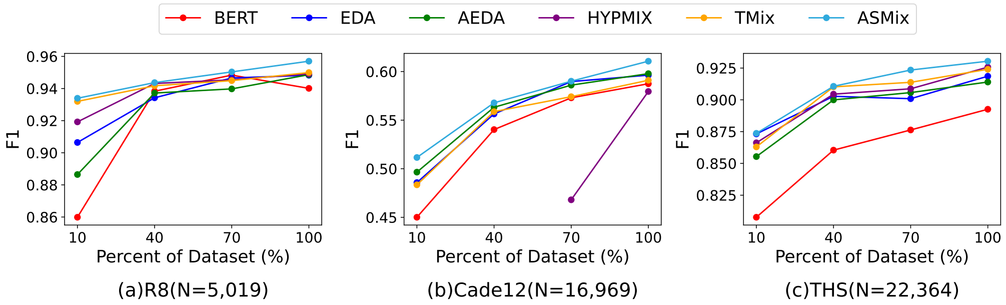
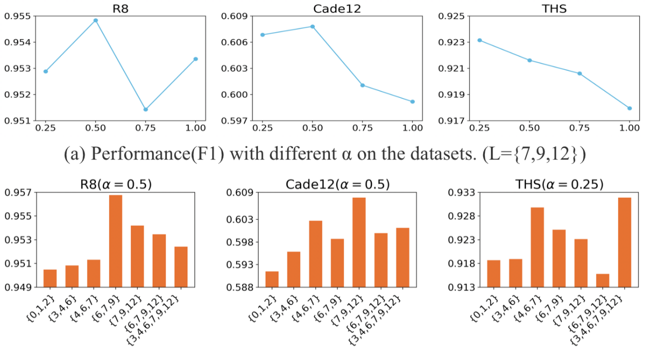


Fig. 3: Performance on benchmark imbalanced text classification tasks for various dataset sizes used for training, compared with different baselines.



(a) Performance(F1) with different α on the datasets. (L={7,9,12})

(b) Performance(F1) with different L on the datasets.

Fig. 4: ASMix on Different Parameters α and L with AMP.

in synthesized samples that are further away from the parent samples.

2) Mixed Layer Set L: Jawahar et al. [19] found that in BERT-based model, $\{3,4,6,7,9,12\}$ are the most informative layers. We chose to mix using different subsets of these layers to see which subset gave the best performance.

Our method achieves the best results on three datasets with $L = \{6,7,9\}$, $L = \{7,9,12\}$, and $L = \{3,4,6,7,9,12\}$, respectively. These layers mainly capture syntactic and semantic features that are very helpful for classification, such as the depth of the syntactic tree, the sequence of top level constituents in the syntax tree, sensitivity to word order, and the sensitivity to random replacement of a noun or verb. If we just mixup at the input and lower layers ($\{0,1,2\}$), there seemed no performance increase.

V. Conclusion

To alleviate the data imbalance problem, this work proposed an effective attention-based smooth data augmentation method, ASMix. This method augments the model input by using smoothed representations and varies minority class samples by adding discriminative information from other samples. Extensive experiments on three benchmark imbalanced text classification datasets prove the effectiveness of ASMix. For future directions, we plan to explore the effect of ASMix on semi-supervised tasks.

References

[1] A. Estabrooks, et al., “A multiple resampling method for learning from imbalanced data sets,” *Computational intelligence*, vol. 20, no. 1, pp. 18–36, 2004.

[2] I. Provilkov and A. Malinin, “Multi-sentence resampling: A simple approach to alleviate dataset length bias and beam-search degradation,” in *EMNLP*, 2021.

[3] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in *ICLR*, 2018.

[4] J. Chen, Z. Yang, and D. Yang, “Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification,” in *ACL*, 2020, pp. 2147–2157.

[5] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, “Cutmix: Regularization strategy to train strong classifiers with localizable features,” in *ICCV*, 2019.

[6] X. Wu, C. Gao, M. Lin, L. Zang, and S. Hu, “Text smoothing: Enhance various data augmentation methods on text classification tasks,” in *ACL*, 2022.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *NeurIPS*, vol. 30, 2017.

[8] P. Teisseyre, et al., “Cost-sensitive classifier chains: Selecting low-cost features in multi-label classification,” *Pattern Recognition*, vol. 86, pp. 290–319, 2019.

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *ICCV*, 2017, pp. 2980–2988.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” in *JAIR*, vol. 16, pp. 321–357, 2002.

[11] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, “Unsupervised data augmentation for consistency training,” in *NeurIPS*, 2020.

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

[13] R. Sawhney, M. Thakkar, S. Agarwal, D. Jin, D. Yang, and L. Flek, “Hypmix: Hyperbolic interpolative data augmentation,” in *EMNLP*, 2021, pp. 9858–9868.

[14] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *NAACL*, 2019.

[15] J. Tian, S. Chen, X. Zhang, and Z. Feng, “A graph-based measurement for text imbalance classification,” in *ECAI 2020*. 2020, pp. 2188–2195.

[16] Z. Waseem and D. Hovy, “Hateful symbols or hateful people? predictive features for hate speech detection on twitter,” in *NAACL*, 2016, pp. 88–93.

[17] J. W. Wei and K. Zou, “EDA: easy data augmentation techniques for boosting performance on text classification tasks,” in *EMNLP-IJCNLP*, 2019.

[18] A. Karimi, L. Rossi, and A. Prati, “AEDA: an easier data augmentation technique for text classification,” in *Findings of ACL*, 2021, pp. 2748–2754.

[19] G. Jawahar, B. Sagot, and D. Seddah, “What does bert learn about the structure of language?” in *ACL 2019*, 2019.