# Augmenting NLP models using Latent Feature Interpolations

Amit Jindal\* Manipal Academy of Higher Education Manipal Academy of Higher Education amitj6460gmail.com

**Aniket Didolkar\*** Manipal Academy of Higher Education adidolkar123@gmail.com

> **Ramit Sawhney** NSIT, New Delhi ramits.co@nsit.net.in

**Arijit Ghosh Chowdhury\*** 

arijit10@gmail.com

Di Jin MIT jindi@mit.edu

**Rajiv Ratn Shah** MIDAS, IIIT Delhi rajivratn@iiitd.ac.in

# Abstract

Models with a large number of parameters are prone to over-fitting and often fail to capture the underlying input distribution. We introduce *Emix*, a data augmentation method that uses interpolations of word embeddings and hidden layer representations to construct virtual examples. We show that *Emix* shows significant improvements over previously used interpolation based regularizers and data augmentation techniques. We also demonstrate how our proposed method is more robust to sparsification. We highlight the merits of our proposed methodology by performing thorough quantitative and qualitative assessments.

#### 1 Introduction

In this paper, we deal with the problem of sentence classification. Current state-of-the-art deep learning approaches for sentence classification have millions of parameters, thus requiring a large number of training examples, which may be time-consuming and expensive to obtain. We present a new data augmentation technique called *Emix* to improve the performance of current text classification models.

In general, it is difficult to come up with rules for language transformation similar to image transformations; hence universal data augmentation techniques have not thoroughly been explored in NLP yet (Wei and Zou, 2019). One such technique is Mixup - it uses systematic transformations to make sure the model trains on samples from the vicinity distribution (Chapelle et al., 2001) along with the original distribution of the training data. Mixup is a data-agnostic data augmentation method that is proven to be effective in Computer Vision tasks (Zhang et al., 2017; Verma et al., 2018). Guo et al. (2019) extends Mixup to Text classification using interpolations in the embedding space. In our proposed method, we eliminate the common mean vector from the embeddings. We then interpolate hidden representations at multiple layers, including the embedding layer. This mixing method is also simple and easy to implement and leads to a further improvement in performance.

## **Related Work:**

Data Augmentation has been vastly used in Computer Vision (Krizhevsky et al., 2012), which employs techniques like flipping, scaling, and rotation. Data augmentation has also been explored in NLP to some extent. (Zhang et al., 2015) replaced words with their synonyms according to a geometric distribution. (Wang and Yang, 2015) used k-NN and Cosine Similarity metrics to find a similar word for replacement. (Wei and Zou, 2019) introduced EDA (Easy Data Augmentation) for NLP which includes Synonym Replacement, Random Insertion, Random Deletion, and Random Swap. These methods have the risk of completely changing the context of the sentence with the replacement of a word which alters the entire

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http:// creativecommons.org/licenses/by/4.0/.

meaning of the sentence. We create virtual examples by interpolating the latent spaces of two sentences and mix the corresponding labels accordingly to ensure that the interpolated sentence still conforms to the correct label.

Mixup (Zhang et al., 2017; Verma et al., 2018; Tokozume et al., 2018; Yun et al., 2019) has shown improvements in the accuracy of image classification models in Computer Vision. In NLP, (Guo et al., 2019) trained a CNN based classifier on various sentiment classification datasets with mixup and found an increase in the generalization capabilities of the model. However, this methodology only interpolates features within the input spaces. Through *Emix*, we capture a greater breadth of the feature space by interpolating hidden layer representations.

**Our Contribution:** We make the following contributions : i) We create improved sentence classification models using **Emix**, a technique that uses latent interpolations of hidden layer representations. ii) We demonstrate how Emix makes text classification models more robust to sparsification. iii) Qualitatively, we highlight how *Emix* leads to more pronounced decision boundaries for text classification.

#### 2 Methodology

The main idea of Mixup (Zhang et al., 2017) is that given two labeled data points  $(x_i, y_i)$  and  $(x_j, y_j)$ , where  $x_i$  and  $x_j$  are two random samples and  $y_i$  and  $y_j$  are the one-hot representation of the label. The algorithm creates virtual training samples by linear interpolation of input as well as labels:

$$x = mix(x_i, x_j) = \lambda x_i + (1 - \lambda)x_j \tag{1}$$

$$y = mix(y_i, y_j) = \lambda y_i + (1 - \lambda)y_j \tag{2}$$

where  $\lambda \sim \beta(\alpha, \alpha)$  is the mixing ratio and  $\tilde{y}$  is defined as the mixed label. Mixup was demonstrated to work well on continuous image data. However, extending it to text is challenging since it is infeasible to compute the interpolation of discrete tokens. Here, we propose a new mixing method where the neural network is trained on interpolations of the hidden states and takes into account the properties of word representation. The word embeddings have non-zero mean, word vectors share a large common vector. Since all words share the same common vector we first eliminate them by removing the non-zero mean vector from all word vectors, effectively reducing the energy (Mu et al., 2017). Let  $x_i(w) \in \mathbb{R}^d$  be a word representation for a given word w in the vocabulary V.  $x_i(w)$  shares a non-zero common vector,  $x_i(w) = \tilde{x}_i(w) + \mu_i$ , where  $\mu_i$  is the average of all  $x_i(w)$ 's, i.e.,  $\mu_i = \frac{1}{|V|} \sum_{w \in V} x_i(w)$ .

Second, we take the difference of text energies (Mu et al., 2017) of the samples into consideration analogous to how Tokozume et al. (2017) takes into account the energy of the speech samples, in order to make the perception of the mixed sample  $x_i : x_j = \lambda : (1 - \lambda)$ . We define t using the standard deviation per text sample ( $\sigma_i$  and  $\sigma_j$ ) so that the ratio of text energy becomes  $x_i : x_j = \lambda : (1 - \lambda)$ . We solve  $t\sigma_i :$  $(1 - t) \sigma_j = \lambda : (1 - \lambda)$  and obtain the proposed mixing method:

$$mix(x_{i}, x_{j}) = \frac{t(x_{i} - \mu_{i}) + (1 - t)(x_{j} - \mu_{i})}{\sqrt{t^{2} + (1 - t)^{2}}}$$

$$where \ t = \frac{1}{1 + \frac{\sigma_{i}}{\sigma_{j}} \cdot \frac{1 - \lambda}{\lambda}}$$
(3)

Let  $g(.,\theta)$  denote the classification model used, where  $\theta$  denotes the model parameters. Assuming this model has M layers, we choose to mix the hidden representations at the m-th layer,  $m \in [0, M]$ . Mathematically the m-th layer is denoted as  $g_m(.,\theta)$ , hence the hidden representation of the m-th layer is  $h_m = g_m(h_{m-1}, \theta)$ . The 0-th layer is considered as the embedding layer. Hence, for two text samples  $x_i$  and  $x_j$ ,  $h_0^j = W_E x_i$ ,  $h_0^j = W_E x_j$  and the following hidden representations are as follows:

$$h_{l}^{i} = g_{l}(h_{l-1}^{i}, \theta), l \in [1, m]$$
(4)

$$h_{l}^{j} = g_{l}(h_{l-1}^{j}, \theta), l \in [1, m]$$
(5)

These hidden representations at the *m*-th layer are mixed using equation 3. We denote this mixed representation as  $h_m$ . Mixup at the *m*-th layer is thus defined as follows:

$$h_m = \frac{th_m^i + (1-t)h_m^j}{\sqrt{t^2 + (1-t)^2}} \tag{6}$$

The continued forward pass after the mixed hidden representation has been generated is defined as follows:

$$h_{l} = g_{l}(h_{l-1}, \theta), l \in [m+1, M]$$
(7)

The layers chosen for mixup are denoted by S where  $S = \{S_1, S_2, ...\}$  where each  $S_i \in [0, M]$ . The layer m where mixup occurs is chosen randomly from S with equal probability given to each layer in S and sampled separately for each pair of examples that are mixed.

### **3** Experiments

We investigated the effectiveness of EMix with five benchmark sentence classification tasks as used by Guo et al. (2019), for a fair comparison. The following datasets are used : TREC (Li and Roth, 2002), SST-1 (Socher et al., 2013), SST-2 (binary classification) and Subj (Pang and Lee, 2004).

We evaluate our proposed mixup using the popular CNN (Kim, 2014) and BERT (Devlin et al., 2018) for sentence classification model. We compare with three recent text augmentation methods including EDA (Wei and Zou, 2019), wordMixup (linear interpolation is applied on the word embedding level) and senMixup (linear interpolation is conducted on the layer before the Softmax layer) (Guo et al., 2019). These augmentation strategies rely less on additional text resources or domain knowledge.

In our experiments, we follow the exact implementation and settings in (Kim, 2014), (Guo et al., 2019), and (Wei and Zou, 2019). Specifically, we use filter sizes of 3, 4, and 5, each with 100 feature maps; dropout rate of 0.5 and L2 regularization of 0.2 for the baseline CNN model. We use the HuggingFace (Wolf et al., 2019) implementation of the BERT base model with 12 transformer blocks, 12 attention heads, and 110 *million* parameters. We use the default learning rate of 0.00002, dropout rate as 0.1, and batch size 8 for all experiments with BERT. All models were trained on Nvidia Tesla K80 GPU.

#### 4 Observations

#### **Observation 1: Impact of EMix on Sentence Classification**

Table 1 shows the accuracy for text classification. The results demonstrate how Emix outperforms word-Mixup and senMixup (Guo et al., 2019). Emix leverages interpolations from intermediate layers, which capture richer features to provide an additional training signal.

Wei and Zou (2019) noted that EDA does not work for regularized pretrained models like BERT. We observe the same across our experiments as well, as absolute improvements with EDA are limited to a maximum of 0.5 %. However, we observe substantial absolute improvements with EMix, using CNN as well as BERT. Another point to note is that EDA doesn't show major improvements when the dataset is sufficiently large, as is with the case of UCI News dataset. On the same dataset, Emix shows an absolute improvement of 2.2 % (CNN) and 3.2 % (BERT).

#### **Observation 2: Performance after Sparsification:**

Over parameterized networks have a considerably large memory and environmental footprint. Recent works (Chirkova et al., 2018) have shown that test accuracy similar to a large network can be achieved with only a fraction of it's parameters. We check the impact of Mixup on models with fewer parameters obtained through pruning.

Various pruning techniques have been suggested to approximate weight's importance (LeCun et al., 1990). We test our models with weight pruning, where we mask out weights with magnitude lesser than a calculated threshold value.

We observe that EMix is more robust to pruning at all levels of sparsity (Table 2). This indicates that EMix eliminates noisy weight estimates by providing more consistent gradient estimates that allow weights to be concentrated in some localized regions of the network.



Figure 1: Embeddings of UCI dataset formed by vanilla and EMix BERT model visualized using PCA.

Model	SST-1	SST-2	TREC	SUBJ	UCI News
CNN	$43.6\pm0.19$	$82.3\pm0.47$	$90.2\pm0.20$	$90.6\pm0.45$	$80.23\pm0.25$
CNN + EDA	$45.1\pm0.54$	$83.1\pm0.49$	$89.2\pm0.65$	$90.8\pm0.44$	$80.0\pm0.79$
CNN + wordMixup	$45.2\pm0.90$	$82.8\pm0.45$	$90.9\pm0.42$	$92.9\pm0.41$	$81.0\pm0.39$
CNN + senMixup	$45.2\pm0.22$	$83.0\pm0.35$	$92.1\pm0.31$	$92.7\pm0.38$	$81.9\pm0.76$
CNN + EMix	$\textbf{47.5} \pm \textbf{0.23}$	$\textbf{85.5} \pm \textbf{0.40}$	$\textbf{93.2} \pm \textbf{0.70}$	$\textbf{93.1} \pm \textbf{0.42}$	$\textbf{82.4} \pm \textbf{0.20}$
BERT	$53.1\pm0.42$	$91.5\pm0.37$	$97.6\pm0.22$	$97.8\pm0.56$	$92.5\pm0.36$
BERT + EDA	$53.5\pm0.31$	$91.9\pm0.45$	$98.1\pm0.25$	$98.1\pm0.42$	$92.9\pm0.41$
BERT+wordMixup	$53.7\pm0.24$	$92.1\pm0.64$	$98.4\pm0.45$	$98.6\pm0.35$	$93.3\pm0.31$
BERT + senMixup	$52.8\pm0.48$	$91.8\pm0.48$	$97.8\pm0.24$	$97.1\pm0.29$	$93.3\pm0.31$
BERT + EMix	$\textbf{55.1} \pm \textbf{0.72}$	$\textbf{93.3} \pm \textbf{0.54}$	$\textbf{98.8} \pm \textbf{0.56}$	$\textbf{98.9} \pm \textbf{0.61}$	$\textbf{95.7} \pm \textbf{0.42}$

Table 1: Accuracy (%) obtained on the testing methods. We report mean scores over 10 runs with standard deviations (denoted  $\pm$ ). Best results highlighted in **Bold**.

#### 5 Qualitative Analysis : Class Separability

We use the class-to-class (c2c) separability (Khan et al., 2017) defined by estimating the spread of the with-in class samples (intraclass) compared to the between-class (interclass) ones.

The separability (S) is defined as :

$$S(class1, class2) = \frac{1}{N_{class1}} \sum_{i} \frac{dist_{intra}(f_i)}{dist_{inter}(f_i)}$$
(8)

where distances are calculated in the feature space where each point is a 768 dimensional feature vector  $f_i$  from the final layer in BERT, and  $N_{class1}$  is the number of samples belonging to class1. Here, lower the value of S, higher is the separability. Table 3 shows the S values for EMix and vanilla BERT. We see that the average of S value is lower for EMix as compared to vanilla BERT. Figure 1 shows that

Sparsity	BERT	Emix
25%	91.67	93.40
33%	91.76	93.23
50%	89.7	92.87
75%	25.00	82.61

Model	Ent.	Buss.	Tech.	Heal.	Avg
BERT	0.11	0.23	0.26	0.30	0.23
EMix	0.13	0.22	0.18	0.24	0.19

Table 2: Accuracy (%) of Vanilla and EMix BERT model for different prune percentages on UCI-News dataset.

Table 3: Average Separability(S) of the each class from all the other classes for each variant of Mixup in the UCI-News dataset.

the features learned using EMix lead to higher class separability as compared to vanilla BERT and forms clusters with lesser inter-class overlap than vanilla model. Also, from figure 1 we can see the Health and Technology class embeddings are better separated using EMix which were previously highly overlapped in the vanilla model.

## 6 Conclusion

We introduced Emix, a novel interpolation based mixing strategy for text classification that uses linear interpolations of hidden states. We performed thorough quantitative assessments to demonstrate how Emix outperforms strong augmentation baselines, both in terms of accuracy and robustness to weight pruning. Qualitatively, we also elucidate how Emix allows for better separation between classes.

# References

- Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. 2001. Vicinal risk minimization. In Advances in neural information processing systems, pages 416–422.
- Nadezhda Chirkova, Ekaterina Lobacheva, and Dmitry Vetrov. 2018. Bayesian compression for natural language processing. *arXiv preprint arXiv:1810.10927*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.
- Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.
- Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. 2017. Costsensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- Yann LeCun, John S Denker, and Sara A Solla. 1990. Optimal brain damage. In Advances in neural information processing systems, pages 598–605.
- Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.
- Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2017. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*.
- Bo Pang and Lillian Lee. 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*, page 271. Association for Computational Linguistics.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. 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, Seattle, Washington, USA, October. Association for Computational Linguistics.
- Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. 2017. Learning from between-class examples for deep sound recognition. *arXiv preprint arXiv:1711.10282*.
- Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5486–5494.
- Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. 2018. Manifold mixup: Better representations by interpolating hidden states. *arXiv* preprint arXiv:1806.05236.

- William Yang Wang and Diyi Yang. 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*, pages 2557–2563, Lisbon, Portugal, September. Association for Computational Linguistics.
- Jason W Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-theart natural language processing. *ArXiv*, abs/1910.03771.
- Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.
- Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.