MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification

(2020)

Jiaao Chen, Zichao Yang, Diyi Yang

Summary

Contributions

In this paper, the authors propose a novel augmentation technique for text classification based on mixup called TMix. TMix takes as input two text instances, first produces their hidden representation at a given layer, and then interpolates them in their corresponding hidden space. Given that the interpolation is continuous, it gives TMix the potential to create an infinite amount of augmented data samples and thus drastically reducing overfitting in low data regimes. Second, the authors apply the proposed augmentation technique in a semi-supervised setup with an entropy minimization term and demonstrate good performances.

Method

TMix

First, let's introduce the standard augmentation technique mixup. Given two data points $(\mathbf{x}_i, \mathbf{y}_i)$ and $(\mathbf{x}_j, \mathbf{y}_j)$, using mixup we can create virtual training samples by interpolating both the inputs and the labels of the two samples.

$$\tilde{\mathbf{x}} = \text{mix}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j$$
$$\tilde{\mathbf{y}} = \text{mix}\left(\mathbf{y}_i, \mathbf{y}_j\right) = \lambda \mathbf{y}_i + (1-\lambda)\mathbf{y}_j$$

where $\lambda \in [0, 1]$. The objective in this paper is to propose a new augmentation but for textual data, but given the discrete nature of the input tokens, applying such an interpolation at the input space is infeasible. To this end the authors propose TMix where the interpolation is applied in the hidden space rather than the input space. Consider BERT as the model of choise with L layers, the first step is to chose an interpolation layer $m \in [0, L]$, and then produce the hidden representation at this layer by a standard forward pass through the encoder g over all the layers $l \in [1, m]$.

$$\mathbf{h}_l^i = g_l\left(\mathbf{h}_{l-1}^i; \boldsymbol{\theta}\right), l \in [1, m]$$
$$\mathbf{h}_l^j = g_l\left(\mathbf{h}_{l-1}^j; \boldsymbol{\theta}\right), l \in [1, m]$$

Then given the two hidden representations of the two inputs at layer m, we can apply the same mixup operation resulting in a new mixed hidden state, which is then forwarded through the rest of the layers $[m+1, L]$

$$\tilde{\mathbf{h}}_m = \lambda \mathbf{h}_m^i + (1-\lambda)\mathbf{h}_m^j$$
$$\tilde{\mathbf{h}}_l = g_l\left(\tilde{\mathbf{h}}_{l-1}; \boldsymbol{\theta}\right), l \in [m+1, L]$$

And as such, we have a new and augmented output representation h_L which is a result of the TMix operation. To summarize, the TMix operation is defined as follows.

$$\text{TMix}\left(\mathbf{x}_i, \mathbf{x}_j; g(.; \boldsymbol{\theta}), \lambda, m\right) = \tilde{\mathbf{h}}_L$$

Then during training, an additional TMix loss can be used where we force the model to have similar output over the mixed hidden state and the mixed labels.

$$L_{\text{TMix}} = \text{KL}\left(\text{mix}\left(\mathbf{y}_i, \mathbf{y}_j\right) \| p\left(\text{TMix}\left(\mathbf{x}_i, \mathbf{x}_j\right); \phi\right)\right)$$

MixText: TMix for Semi-supervised learning.

The second step is to integrate the proposed method into a semi-supervised setting. In a semi-supervised setup, instead of having a single labeled set, we have two set, a small labeled set with inputs $\mathbf{X}_l$ a labels $\mathbf{Y}_l$, and a larger unlabeled set $\mathbf{X}_u$. And the objective is to use the unsupervised data to extract additional training signal from the unlabeled examples. In order to utilize the unlabeled data in the TMix loss, we first need to produce the labels of each unlabeled data point. To this end the authors proposed to produce each labels for each data points based on the average prediction over many augmentations. To be more precise, given a input data augmentation such as back-translation, and for a given input sample, we first produce K augmented examples, pass them trough the model resulting in K prediction, sharped the labels to avoid having uniform prediction and then take the average of the K sharpened predictions to have a label of a single unlabeled data point. To summarize, this is done as follows.

$$\begin{aligned}
&1 - \mathbf{x}^a_{i,k} = \text{augment}_k\left(\mathbf{x}^u_i\right), k \in [1, K] \\
&2 - \text{Sharpen}\left(\mathbf{y}^u_i, T\right) = \frac{\left(\mathbf{y}^u_i\right)^{\frac{1}{T}}}{\left\|\left(\mathbf{y}^u_i\right)^{\frac{1}{T}}\right\|_1} \\
&3 - \mathbf{y}^u_i = \frac{1}{w_{\text{ori}} + \sum_k w_k}\left(w_{\text{ori}}\, p\left(\mathbf{x}^u_i\right) + \sum_{k=1}^{K} w_k p\left(\mathbf{x}^a_{i,k}\right)\right)
\end{aligned}$$

As a results, we have two sets, the labeled set $\mathbf{X}_l$, and newly labeled $\mathbf{X}_u$, but also the augmented set of unlabeled data, where we assign the produced labels above to each examples of the K augmentations, resulting in a new set $\mathbf{X}_a$. Finaly, the new training set is the super set of the three sets $\mathbf{X} = \mathbf{X}_l \cup \mathbf{X}_u \cup \mathbf{X}_a$ and their labels $\mathbf{Y} = \mathbf{Y}_l \cup \mathbf{Y}_u \cup \mathbf{Y}_a$. Finally, in addition to the standard cross-entropy loss over the labeled set, we have TMix loss over the supervised, and also the entropy loss to push the model to produce confident prediction which will help with the label prediction process above.

$$\begin{aligned}
L_{\text{MixText}} &= L_{\text{TMix}} + \gamma_m L_{\text{margin}} \\
L_{\text{margin}} &= \mathbb{E}_{\mathbf{x} \in \mathbf{X}_u} \max\left(0, \gamma - \|\mathbf{y}^u\|_2^2\right)
\end{aligned}$$

The whole process is detailed in the figure bellow:

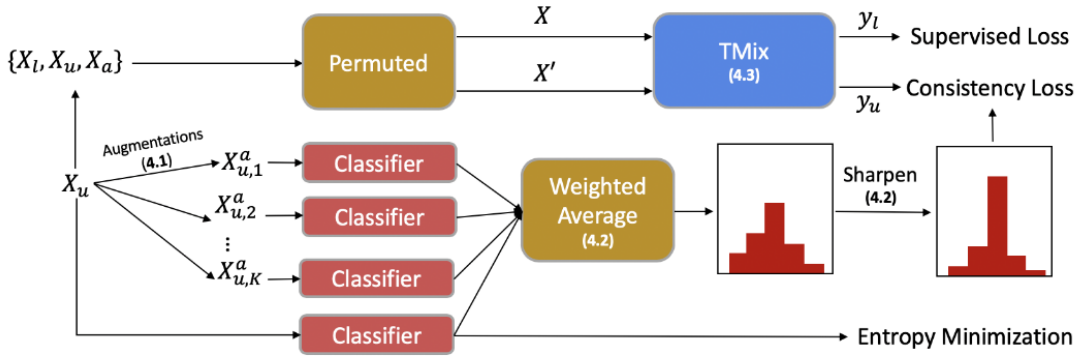

Figure 2: Overall Architecture of MixText. MixText takes in labeled data and unlabeled data, conducts augmentations and predicts labels for unlabeled data, performs TMix over labeled and unlabeled data, and computes supervised loss, consistency loss and entropy minimization term.

Results

Dataset	Label Type	Classes	Unlabeled	Dev	Test
AG News	News Topic	4	5000	2000	1900
DBpedia	Wikipeida Topic	14	5000	2000	5000
Yahoo! Answer	QA Topic	10	5000	5000	6000
IMDB	Review Sentiment	2	5000	2000	12500

Table 1: Dataset statistics and dataset split. The number of unlabeled data, dev data and test data in the table means the number of data per class.

Datset	Model	10	200	2500	Dataset	Model	10	200	2500
AG News	VAMPIRE	-	83.9	86.2	DBpedia	VAMPIRE	-	-	-
	BERT	69.5	87.5	90.8		BERT	95.2	98.5	99.0
	TMix*	74.1	88.1	91.0		TMix*	96.8	98.7	99.0
	UDA	84.4	88.3	91.2		UDA	97.8	98.8	99.1
	MixText*	**88.4**	**89.2**	**91.5**		MixText*	**98.5**	**98.9**	**99.2**
Yahoo!	VAMPIRE	-	59.9	70.2	IMDB	VAMPIRE	-	82.2	85.8
	BERT	56.2	69.3	73.2		BERT	67.5	86.9	89.8
	TMix*	58.6	69.8	73.5		TMix*	69.3	87.4	90.3
	UDA	63.2	70.2	73.6		UDA	78.2	89.1	90.8
	MixText*	**67.6**	**71.3**	**74.1**		MixText*	**78.7**	**89.4**	**91.3**

Table 2: Performance (test accuracy(%)) comparison with baselines. The results are averaged after three runs to show the significance (Dror et al., 2018), each run takes around 5 hours. Models are trained with 10, 200, 2500 labeled data per class. VAMPIRE, Bert, and TMix do not use unlabeled data during training while UDA and MixText utilize unlabeled data. * means our models.

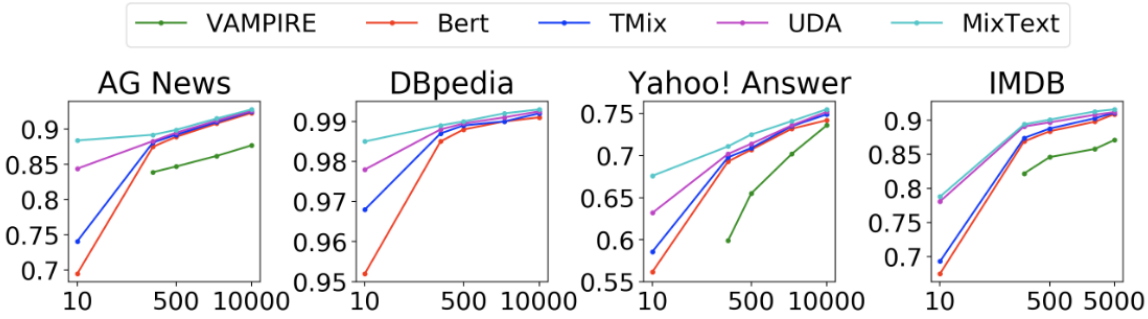

Figure 3: Performance (test accuracy (%)) on AG News, DBpedia, Yahoo! Answer and IMDB with 5000 unlabeled data and varying number of labeled data per class for each model.

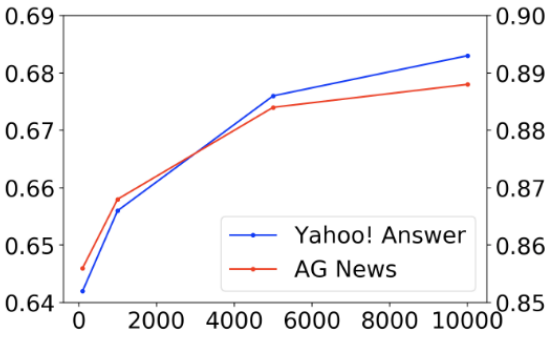

Figure 4: Performance (test accuracy (%)) on AG News (y axis on the right) and Yahoo! Answer (y axis on the left) with 10 labeled data and varying number of unlabeled data per class for MixText.

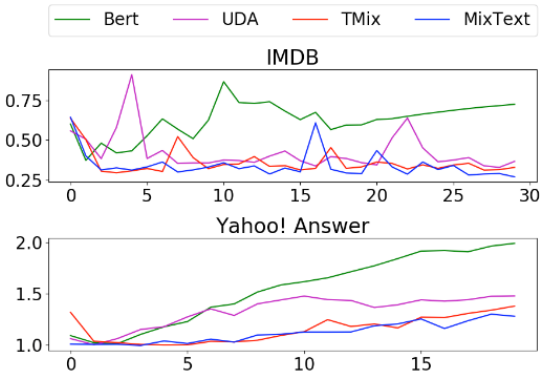

Figure 5: Loss on development set on IMDB and Yahoo! Answer in each epoch while training with 200 labeled data and 5000 unlabeled data per class.

Mixup Layers Set	**Accuracy(%)**
$\emptyset$	69.5
{0,1,2}	69.3
{3,4}	70.4
{6,7,9}	71.9
{7,9,12}	**74.1**
{6,7,9,12}	72.2
{3,4,6,7,9,12}	71.6

Table 3: Performance (test accuracy (%)) on AG News with 10 labeled data per class with different mixup layers set for TMix. $\emptyset$ means no mixup.

Model	**Accuracy(%)**
MixText	**67.6**
- weighted average	67.1
- TMix	63.5
- unlabeled data	58.6
- all	56.2

Table 4: Performance (test accuracy (%)) on Yahoo! Answer with 10 labeled data and 5000 unlabeled data per class after removing different parts of MixText.