A Data Cartograph based Mix-Up for Pre-trained Language Models

-Summary-

Introduction

Background

- Using some scores or metrics, we can evaluate difficulty or consistency of examples
- Well-known characterization of data :
  1. Easy-to-learn : samples that model predicts correctly and consistently
  2. Ambiguous : samples where true class probabilities vary frequently during training
  3. Hard-to-learn : samples that are potentially mis-labeled or erroneous
- It turns out that easy-to-learn samples are useful for model optimization and help model to converge, while ambiguous samples are the most beneficial for learning (due to reasonable difficulty of the sample)

- One idea using mix-up : mix-up easy-to-learn and ambiguous samples to make samples potentially helpful for learning.

TDMixUp

Proposed approach : TDMixUp

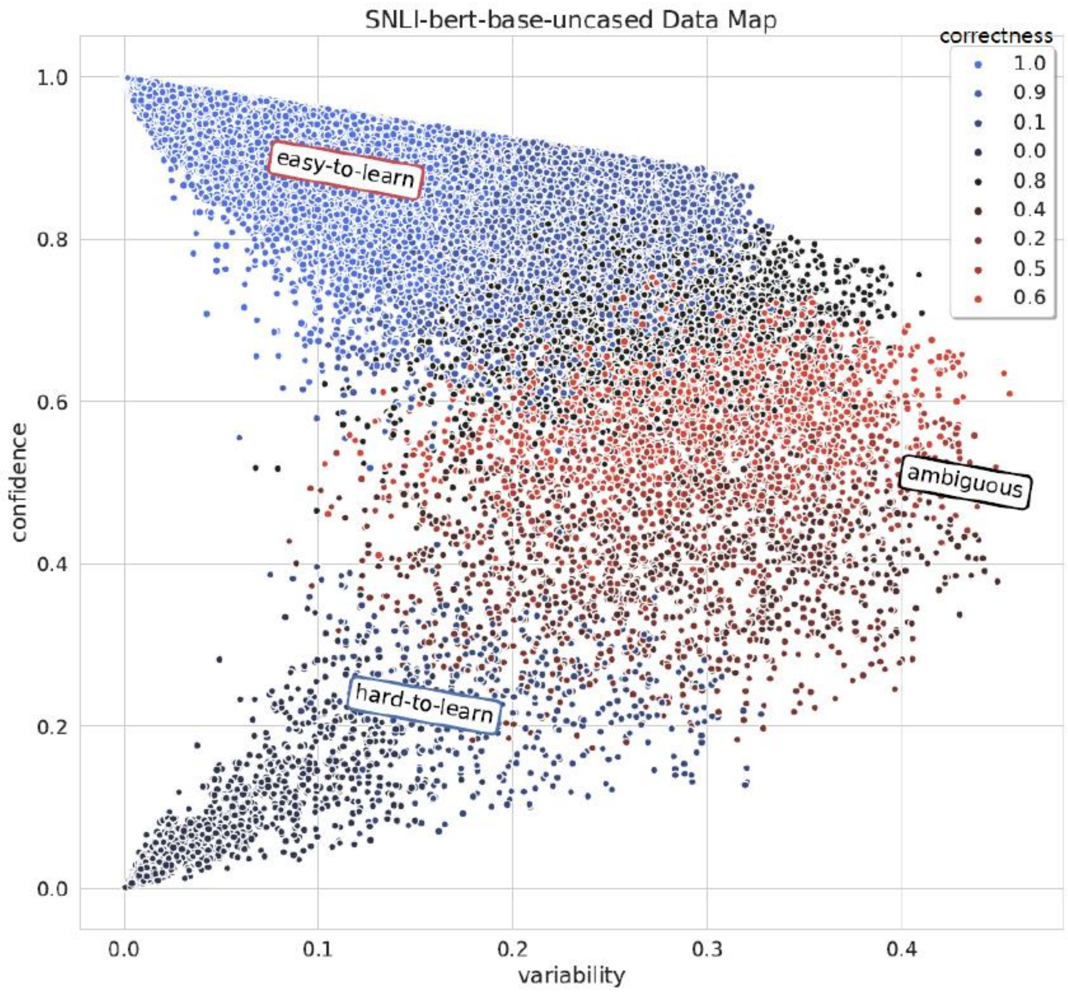
- Method : generated additional samples based on the characteristics of the data samples
  1. Reveal the characteristics of each data sample by using training dynamics (i.e : confidence, variability, and Area Under the Margin (AUM))
  2. Generate samples by applying mix-up between easy-to-learn and ambiguous samples

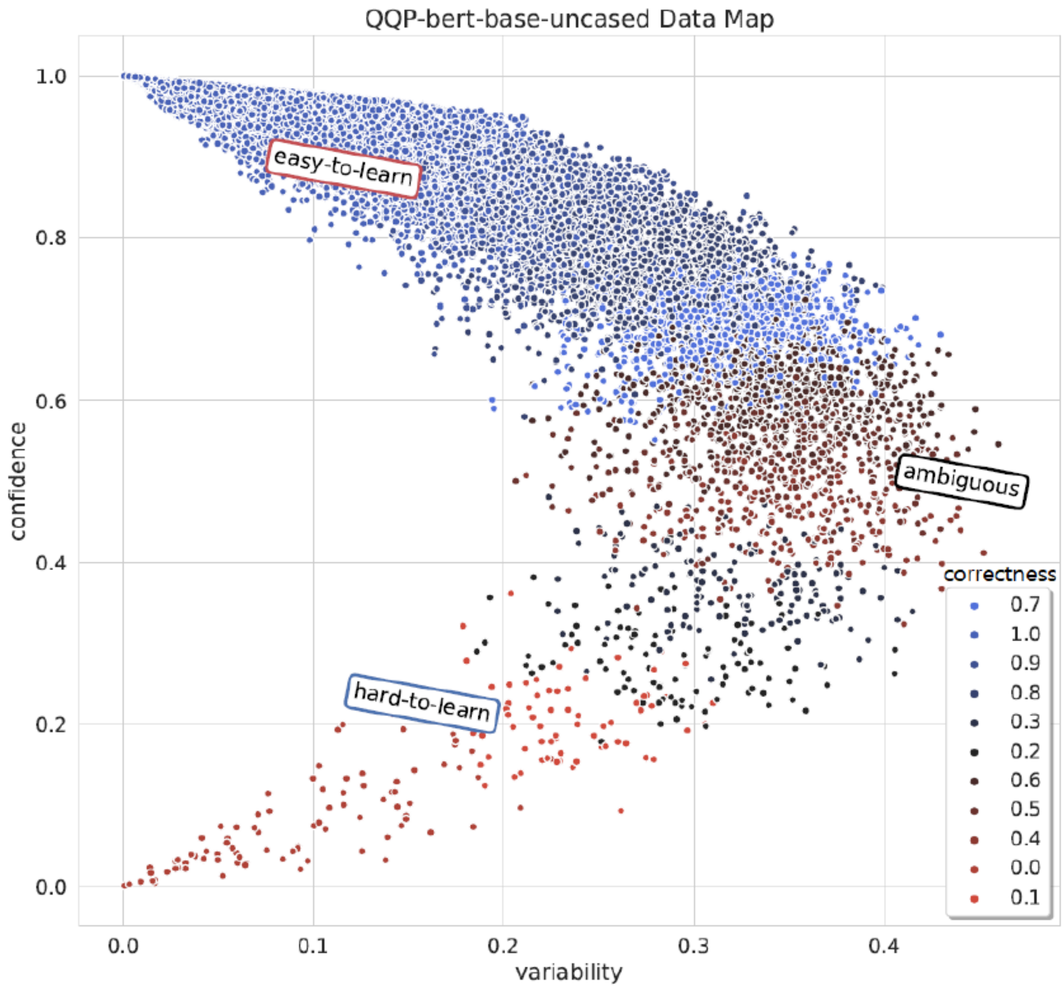
Statistics are calculated for each sample (x_i, y_i) over E training epoch

- Confidence : $\hat{\mu}_i = \frac{1}{E}\sum_{e=1}^{E} p_{\theta^{(c)}}(y_i|x_i)$, where $p_{\theta^{(c)}}$: model's probability with parameter $\theta^{(e)}$ at the end of eth epoch [mean model probability of the true label y_i across epochs]

- Variability : $\widehat{\sigma_i} = \sqrt{\frac{\sum_{e=1}^{E}\left(p_{\theta^{(e)}}(y_i|x_i) - \hat{\mu}_i\right)^2}{E}}$ [standard deviation of $p_{\theta^{(e)}}$ across epochs E]

TDMixUp





Data Maps of SNLI, QQP on BERT-base-uncased model

TDMixUp

Area Under the Margin (AUM) [Pleiss, 2020]

- AUM measures how different a true label for a sample is compared to a model's belief at each epoch
- We compute $AUM(x_i, y_i)$ as the area under the margin averaged across all training epochs E.
- Margin of (x_i, y_i) at epoch e : $M^e(x_i, y_i) = z_{y_i} - \max_{y_i \neq k}(z_k)$

  where z_{y_i} is the logit of x_i corresponding to the true label y_i
- AUM of (x_i, y_i) : $AUM(x_i, y_i) = \frac{1}{E}\sum_{e=1}^{E} M^e(x_i, y_i)$
- Contrast to confidence, AUM measures how much the top-1 label logit value differs from the other largest logit value, which allows identifying mis-labeled samples.
- How to identify mis-labeled samples?
  1. Train fake data (threshold samples) and calculate AUM of those data
  2. Data with similar or worse AUMs than threshold samples are assumed to be mis-labeled (pick kth percentile AUM of threshold samples as the 'threshold AUM')

Experiments and Results

Tasks and Datasets

- Evaluate TDMixUp on Natural Language Inference(NLI) / Paraphrase Detection / Commonsense Reasoning tasks
- Natural Language Inference :
  - ✓ Dataset = SNLI(Stanford NLI) / MLNI (Multi-genre LNI)
  - ✓ task = to predict if the relation between a hypothesis and a premise is ‘entailment’, ‘contradiction’ or ‘neutral’
- Paraphrase Detection :
  - ✓ Dataset = QQP(Quora Question Pairs) / TPPDB(TwitterPPDB)
  - ✓ task = to test if two questions are semantically equivalent
- Commonsense Reasoning :
  - ✓ Dataset = SWAG (Situations With Adversarial Generations) / HellaSWAG
  - ✓ task = to choose the most plausible continuation of a sentence among four candidates
- Model : BERT based classification model

Experiments and Results

	SNLI		QQP		SWAG	
	Acc	ECE	Acc	ECE	Acc	ECE
100% train	**90.04**$_{0.3}$	2.54$_{0.8}$	**90.27**$_{0.3}$	2.71$_{0.5}$	**79.40**$_{0.4}$	2.49$_{1.8}$
33% train, Easy-to-learn	82.78$_{0.6}$	16.22$_{0.7}$	63.16$_{0.1}$	36.88$_{0.1}$	75.39$_{0.2}$	17.51$_{0.1}$
24% train, Easy-to-learn with AUM	83.03$_{0.9}$	15.05$_{0.9}$	66.43$_{0.6}$	33.93$_{0.8}$	75.56$_{0.1}$	15.81$_{0.7}$
33% train, Ambiguous	89.71$_{0.5}$	**0.74**$_{0.1}$	87.51$_{0.5}$	1.71$_{0.4}$	75.91$_{0.6}$	**1.84**$_{0.7}$
24% train, Ambiguous with AUM	87.88$_{0.7}$	7.09$_{0.8}$	88.63$_{0.5}$	6.36$_{0.6}$	71.74$_{0.4}$	7.55$_{1.1}$
66% train, Easy-to-learn & Ambiguous	89.65$_{0.2}$	2.64$_{0.5}$	90.23$_{0.7}$	**1.35**$_{0.4}$	78.78$_{0.5}$	2.51$_{0.8}$

	MNLI		TwitterPPDB		HellaSWAG	
	Acc	ECE	Acc	ECE	Acc	ECE
100% train	73.52$_{0.3}$	7.09$_{2.1}$	**87.63**$_{0.4}$	8.51$_{0.6}$	**34.48**$_{0.2}$	12.62$_{2.8}$
33% train, Easy-to-learn	61.41$_{0.8}$	36.68$_{1.9}$	81.07$_{0.8}$	18.92$_{0.7}$	33.59$_{1.1}$	29.38$_{2.1}$
24% train, Easy-to-learn with AUM	62.97$_{1.5}$	32.48$_{2.9}$	82.16$_{0.7}$	17.46$_{1.0}$	33.67$_{1.4}$	16.89$_{2.6}$
33% train, Ambiguous	72.52$_{1.2}$	10.73$_{1.0}$	86.62$_{0.6}$	**6.01**$_{1.1}$	34.29$_{0.9}$	8.40$_{1.3}$
24% train, Ambiguous with AUM	70.87$_{0.9}$	17.23$_{1.6}$	86.59$_{0.8}$	7.31$_{0.8}$	33.81$_{1.0}$	**3.76**$_{2.3}$
66% train, Easy-to-learn & Ambiguous	**73.89**$_{0.6}$	**3.46**$_{1.9}$	87.29$_{0.3}$	8.04$_{0.7}$	34.43$_{0.2}$	9.68$_{1.1}$

Comparison of accuracy and expected calibration error (ECE) for several datasets

Experiments and Results

	SNLI		QQP		SWAG	
	Acc	ECE	Acc	ECE	Acc	ECE
100% train	$90.04_{0.3}$	$2.54_{0.8}$	$90.27_{0.3}$	$2.71_{0.5}$	$79.40_{0.4}$	$2.49_{1.8}$
100% train, MixUp (Zhang et al., 2018)	$88.82_{0.2}$	$7.73_{1.1}$	$89.12_{0.5}$	$9.04_{0.8}$	$74.98_{2.3}$	$7.08_{1.0}$
100% train, M-MixUp (Verma et al., 2019)	$89.45_{0.9}$	$1.51_{0.8}$	$89.93_{0.6}$	$3.02_{1.0}$	$78.26_{0.4}$	$4.12_{0.6}$
100% train, MixUp for Calibration (Kong et al., 2020)	$89.25_{0.5}$	$2.16_{0.5}$	$90.24_{0.3}$	$5.22_{0.6}$	$79.44_{0.6}$	$\mathbf{1.10}_{0.4}$
100% train, Back Translation Data Augmentation (Edunov et al., 2018)	$89.22_{0.5}$	$1.98_{0.6}$	$89.18_{0.6}$	$5.01_{0.3}$	$76.22_{0.9}$	$1.24_{0.2}$
66% train, TDMixUp, Easy-to-learn + Ambiguous	$89.73_{0.1}$	$2.39_{0.8}$	$89.77_{0.2}$	$1.89_{0.4}$	$78.38_{0.3}$	$4.21_{0.3}$
57% train, TDMixUp, Easy-to-lean with AUM + Ambiguous (Ours)	$\mathbf{90.31}_{0.2}$	$\mathbf{1.22}_{0.4}$	$\mathbf{90.42}_{0.2}$	$\mathbf{1.53}_{0.9}$	$\mathbf{79.59}_{0.3}$	$2.16_{0.4}$

	MNLI		TwitterPPDB		HellaSWAG	
	Acc	ECE	Acc	ECE	Acc	ECE
100% train	$73.52_{0.3}$	$7.09_{2.1}$	$87.63_{0.4}$	$8.51_{0.6}$	$34.48_{0.2}$	$12.62_{2.8}$
100% train, MixUp (Zhang et al., 2018)	$69.19_{0.8}$	$19.51_{2.1}$	$87.45_{0.3}$	$11.70_{1.6}$	$33.22_{0.4}$	$10.93_{2.0}$
100% train, M-MixUp (Verma et al., 2019)	$73.22_{0.6}$	$8.06_{1.2}$	$87.58_{0.7}$	$7.68_{1.3}$	$34.86_{0.9}$	$13.56_{1.6}$
100% train, MixUp for Calibration (Kong et al., 2020)	$64.90_{0.5}$	$17.75_{1.8}$	$74.51_{1.1}$	$11.83_{1.0}$	$32.51_{0.8}$	$31.61_{2.3}$
100% train, Back Translation Data Augmentation (Edunov et al., 2018)	$73.15_{0.7}$	$8.46_{1.3}$	$86.82_{0.7}$	$8.83_{0.6}$	$34.97_{0.4}$	$22.68_{3.3}$
66% train, TDMixUp, Easy-to-learn + Ambiguous	$72.83_{1.1}$	$5.84_{1.9}$	$87.63_{0.2}$	$6.48_{0.7}$	$34.11_{0.1}$	$10.54_{1.6}$
57% train, TDMixUp, Easy-to-learn with AUM + Ambiguous (Ours)	$\mathbf{74.28}_{0.6}$	$\mathbf{2.91}_{1.4}$	$\mathbf{87.89}_{0.3}$	$\mathbf{6.08}_{0.4}$	$\mathbf{35.21}_{0.6}$	$\mathbf{9.45}_{1.3}$

Accuracy and ECE for several datasets

Experiments and Results

	Acc	ECE	Acc	ECE	Acc	ECE
	SNLI		QQP		SWAG	
Random	89.59	1.70	89.87	3.06	79.15	4.51
Ours	90.31	1.22	90.42	1.53	79.59	2.16
	MNLI		TwitterPPDB		HellaSWAG	
Random	73.22	6.89	87.23	6.53	34.43	15.87
Ours	74.28	2.91	87.89	6.08	35.21	9.45

Accuracy and ECE of Mix-Up selecting random samples on the union of the top 33% easy-to-learn and the top 33% ambiguous samples (Random) and TDMix-Up