Unsupervised Data Augmentation for Consistency Training

Qizhe Xie , Zihang Dai1, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le

Google Brain, Carnegie Mellon University

April 29, 2019

Presented by Fatir Qureshi

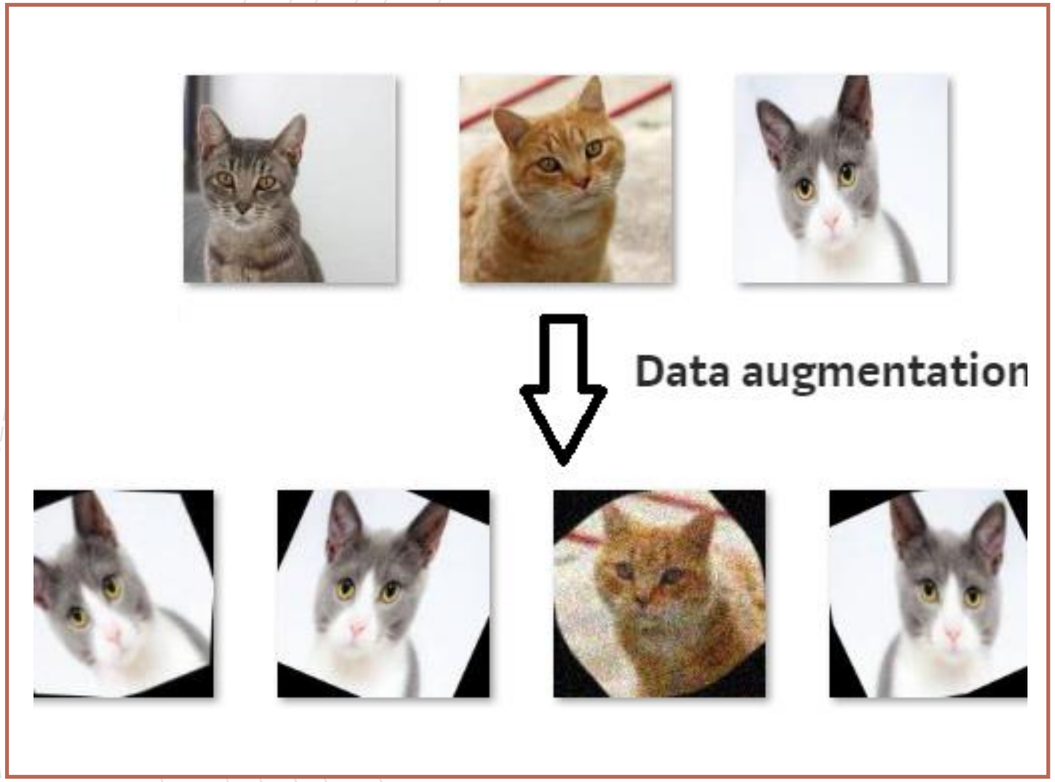
Overview

- Background
- Unsupervised Data Augmentation (UDA) Approach
- Additional Training Techniques
- Experiments and Results
- Conclusions

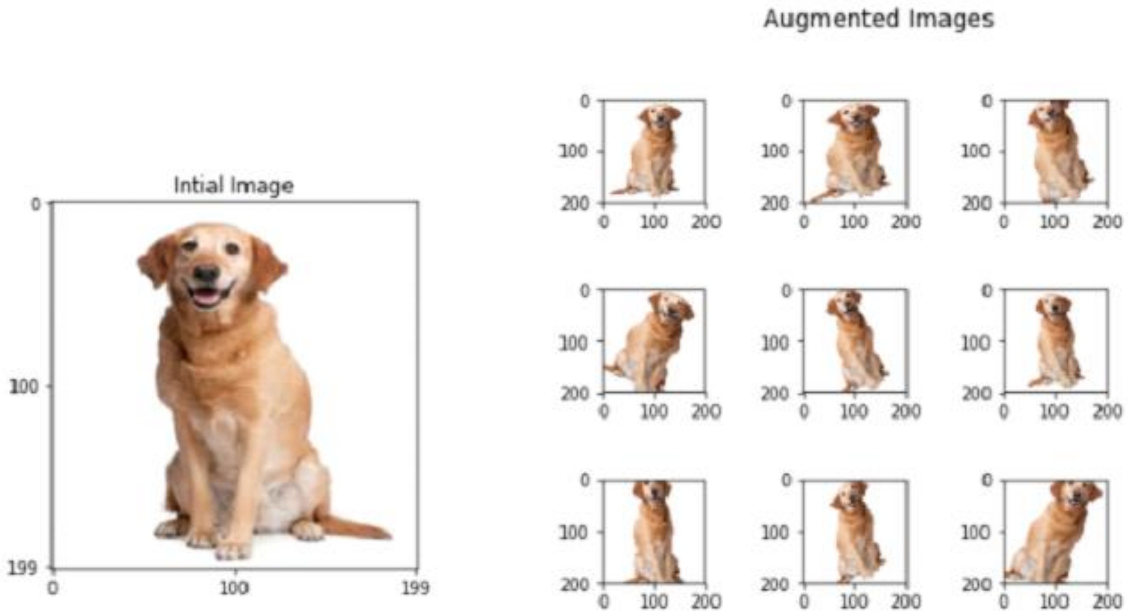
Background

- Extensive amounts of labeled data is typically required for many prominent deep learning techniques
- Efficient methods for using unlabeled data addresses the challenge of acquiring large amounts of labeled data
- Three main approaches exist to semi-supervised learning
  - Graph-based label propagation via graph convolution
  - Modeling prediction target as latent variables
  - Consistency /smoothness enforcing
- Given an observed example, smoothness enforcing methods first create a perturbed version of it, then they enforce the model predictions on the two examples to be similar
- Data augmentation has been promising in alleviating the need for large amounts of labeled data, but has mostly achieved limited gains in supervised settings

Background



- Unsupervised Data Augmentation (UDA) proposes to provide a method by which state-of-the-art data augmentation methods can be applied on unsupervised data
- Method seeks to minimizes the Kullback–Leibler divergence between model predictions on the original example and an example generated by data augmentation



Supervised Data Augmentation

- Goal of data augmentation is to create novel and realistic-looking training data, which is produced by applying a transformation to an example
- Supervised data augmentation can be equivalently seen as constructing an augmented labeled set from the original supervised set and then training the model on the augmented set
- Data augmentation has shown significant promise for NLP, vision and speech

Unsupervised Data Augmentation

One of the main approaches to semi-supervised learning has been enforcing smoothness of the model

Given an input x, compute the output distribution pθ(y | x) given x and a perturbed version pθ(y | x, e) by injecting a small noise e. The noise can be applied to x or hidden states or be used to change the computation process.

Minimize a divergence metric between the two predicted distributions $\mathcal{D}\left(p_\theta(y \mid x) \parallel p_\theta(y \mid x, \epsilon)\right)$

Unsupervised Data Augmentation

- Proposal seeks to use state-of-the-art data augmentation targeted at different tasks as a particular form of perturbation and optimize the same smoothness or consistency enforcing objective on unlabeled examples
- Following virtual adversarial training, minimize the Kullback–Leibler divergence between the predicted distributions on an unlabeled example and an augmented unlabeled example

$$\min_{\theta} \mathcal{J}_{\mathrm{UDA}}(\theta) = \mathop{\mathbb{E}}_{x \in U} \mathop{\mathbb{E}}_{\hat{x} \sim q(\hat{x}|x)} \left[\mathcal{D}_{\mathrm{KL}} \left(p_{\tilde{\theta}}(y \mid x) \,\|\, p_{\theta}(y \mid \hat{x}) \right) \right]$$

$\tilde{\theta}$ is a *fixed* copy of the current parameters

$q(\hat{x} \mid x)$ is a data augmentation transformation

Unsupervised Data Augmentation

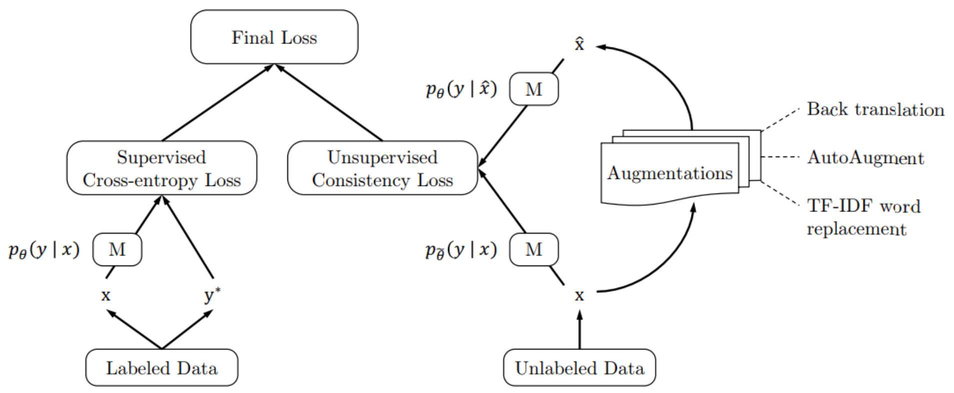
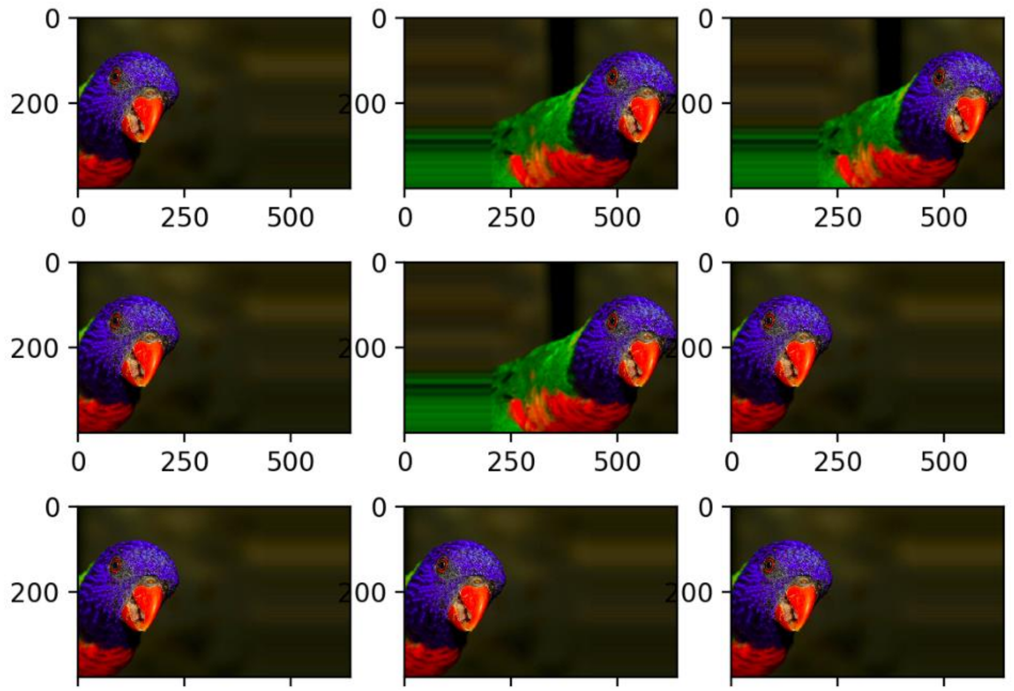


Figure 1: Training objective for UDA, where M is a model that predicts a distribution of y given x

- The objective is defined as:

$$\min_{\theta} \mathcal{J} = \mathop{\mathbb{E}}_{x, y^* \in L} [p_\theta(y^* \mid x)] + \lambda \mathcal{J}_{\text{UDA}}(\theta)$$

- By minimizing the consistency loss, UDA allows for label information to propagate from labeled examples to unlabeled one



Unsupervised Data Augmentation

- Targeted data augmentation as the perturbation function has several advantages
  - Valid perturbation
  - Diverse perturbations
  - Targeted inductive biases

Augmentation Strategies for Different Tasks

- UDA strategy can be applied for multiple different tasks
  - AutoAugment for Image Classification
  - Back translation for Text Classification
  - Term frequency–inverse document frequency-based word replacing for Text Classification

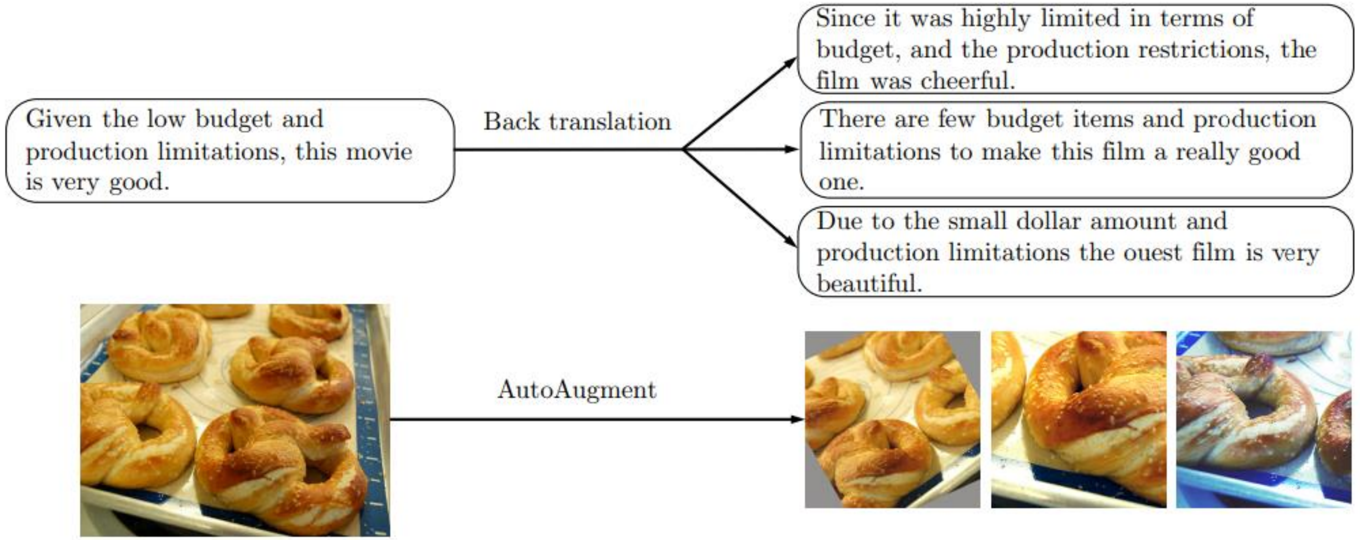


Figure 2: Augmented examples using back translation and AutoAugment

Trade-off Between Diversity and Validity for Data Augmentation

- There is a trade-off between diversity and validity since diversity is achieved by changing a part of the original example, naturally leading to the risk of altering the ground-truth label
- AutoAugment automatically finds the sweet spot between diversity and validity since it is optimized according to the validation set performances in the supervised setting

Training Signal Annealing

- Easier to obtain unlabeled data relative to labeled data - there is a large gap between the amount of unlabeled data and that of labeled data
- To take advantage of as much unlabeled data as possible, a large enough model is needed, but a large model can easily overfit the supervised data of a limited size
- Training Signal Annealing (TSA) is utilized to address this problem

set a threshold $\frac{1}{K} \leq \eta_t \leq 1$, with K being the number of categories

$$\min_{\theta} \frac{1}{Z} \sum_{x, y^* \in B} \left[-I(p_\theta(y^* \mid x) < \eta_t) \log p_\theta(y^* \mid x) \right]$$

- To account for different ratios of unlabeled data and labeled data, consider three particular schedules for ηt :
  - Log- schedule

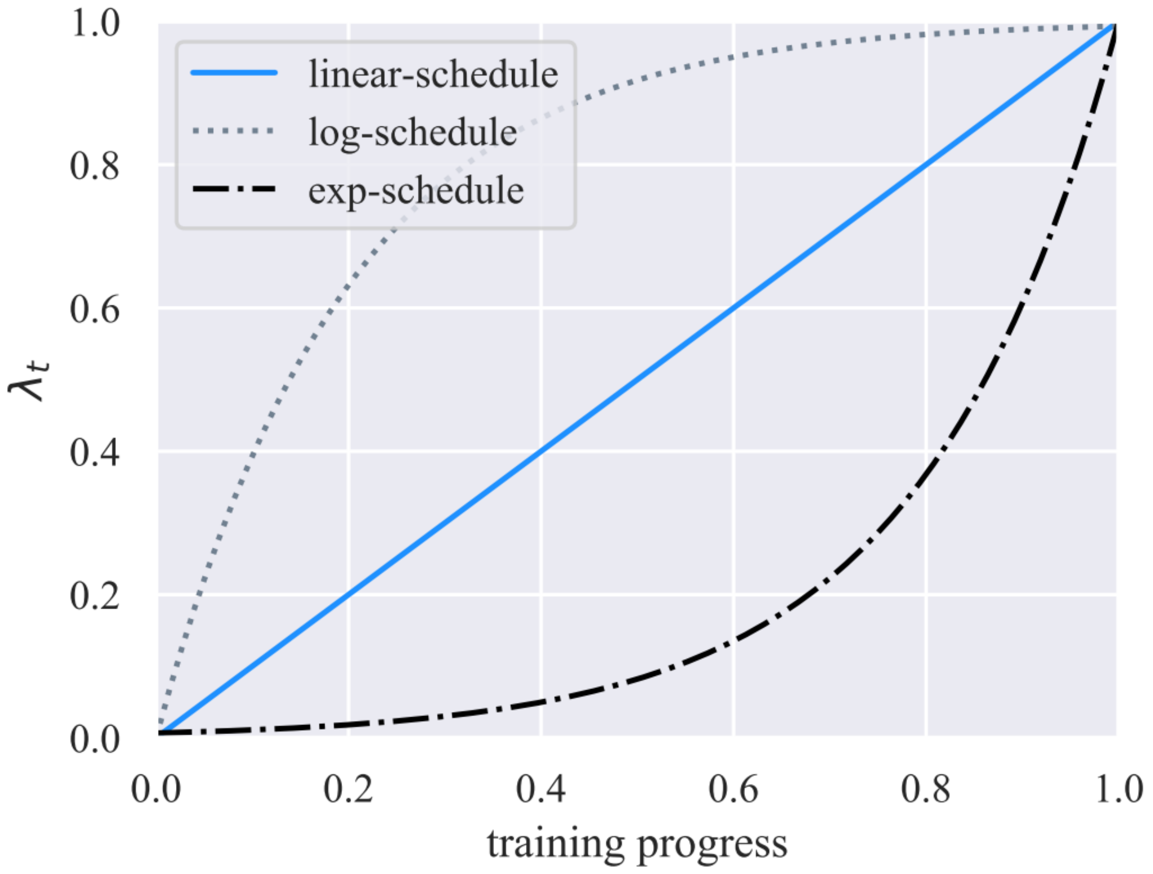
$$\eta_t = (1 - \exp(-\frac{t}{T} * 5)) * (1 - \frac{1}{K}) + \frac{1}{K}$$

  - Linear schedule

$$\eta_t = \frac{t}{T} * (1 - \frac{1}{K}) + \frac{1}{K}$$

  - Exp - schedule

$$\eta_t = \exp((\frac{t}{T} - 1) * 5) * (1 - \frac{1}{K}) + \frac{1}{K}$$



Training Schedule

- When the model is prone to overfit, e.g., when the problem is relatively easy or the number of labeled examples is very limited, the exp-schedule is the most suitable
- When the model is less likely to overfit (e.g., when we have abundant labeled examples or when the model employs effective regularizations), the log-schedule can serve well

Sharpening Predictions

- Confidence-based masking
- Entropy minimization
- Softmax temperature controlling

Experiments

- Utilize UDA with several language, text and vision tasks to assess general performance
- Compare UDA with other semi-supervised learning methods on standard vision benchmarks, CIFAR-10 and SVHN
- Evaluate UDA on ImageNet

Text Classification Experiments

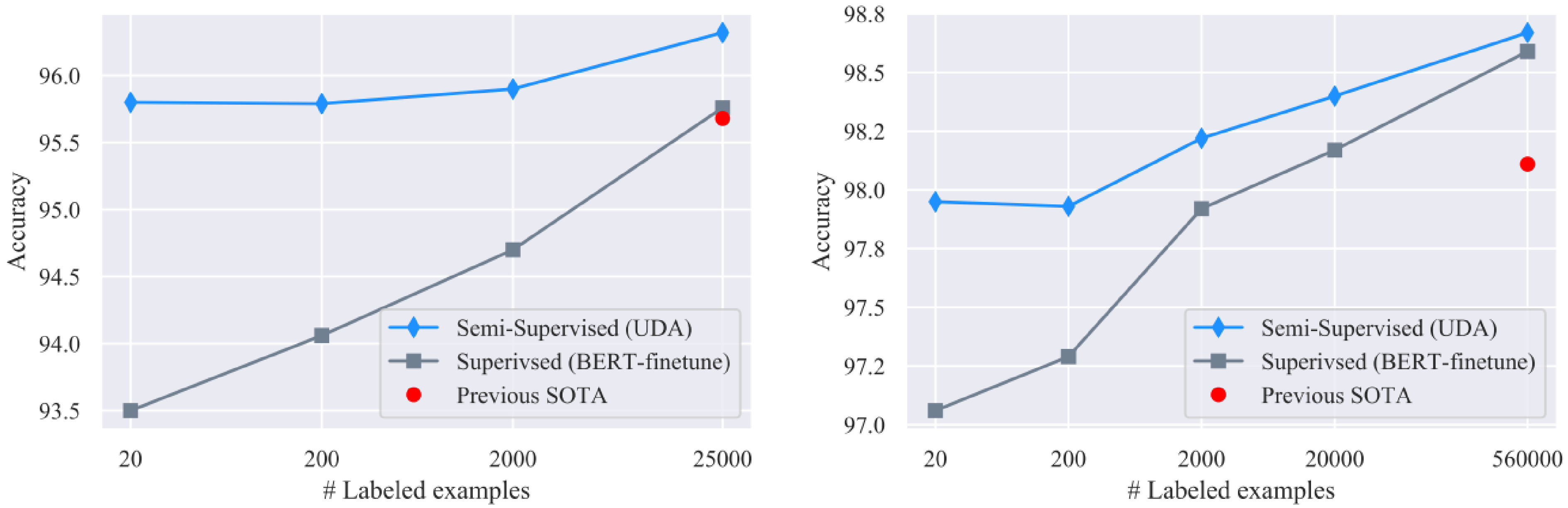
- Carried out experiments on six language datasets, which were IMDb, Yelp-2, Yelp-5, Amazon-2, Amazon-5 and DBPedia
  - DBPedia for category classification
  - Other datasets for sentiment classifications on different demains
- Adopted the Transformer model used in BERT as baseline model due to its great performances on many tasks
- Four initialization schemes considered
  - Random Transformer
  - $\text{BERT}_{\text{Base}}$
  - $\text{BERT}_{\text{Large}}$
  - $\text{BERT}_{\text{FINETUNE}}$: $\text{BERT}_{\text{LARGE}}$
- 20 supervised examples for binary sentiment classification tasks were utilized, and 500 per class for 5 way classification

Main results for Text

Fully supervised baseline							
Datasets (# Sup examples)		IMDb (25k)	Yelp-2 (560k)	Yelp-5 (650k)	Amazon-2 (3.6m)	Amazon-5 (3m)	DBpedia (560k)
Pre-BERT SOTA		*4.32*	2.16	29.98	3.32	34.81	0.70
$\text{BERT}_{\text{LARGE}}$		4.51	*1.89*	*29.32*	*2.63*	*34.17*	*0.64*
Semi-supervised setting							
Initialization	**UDA**	IMDb (20)	Yelp-2 (20)	Yelp-5 (2.5k)	Amazon-2 (20)	Amazon-5 (2.5k)	DBpedia (140)
Random	✗	43.27	40.25	50.80	45.39	55.70	41.14
	✓	25.23	8.33	41.35	16.16	44.19	7.24
$\text{BERT}_{\text{BASE}}$	✗	27.56	13.60	41.00	26.75	44.09	2.58
	✓	5.45	2.61	33.80	3.96	38.40	1.33
$\text{BERT}_{\text{LARGE}}$	✗	11.72	10.55	38.90	15.54	42.30	1.68
	✓	4.78	2.50	33.54	3.93	37.80	1.09
$\text{BERT}_{\text{FINETUNE}}$	✗	6.50	2.94	32.39	12.17	37.32	-
	✓	**4.20**	**2.05**	**32.08**	**3.50**	**37.12**	-

- UDA consistently improves the performance regardless of the model initialization scheme
- With a significantly smaller amount of supervised examples, UDA can offer decent or even competitive performances compared to the SOTA model trained with full supervised data

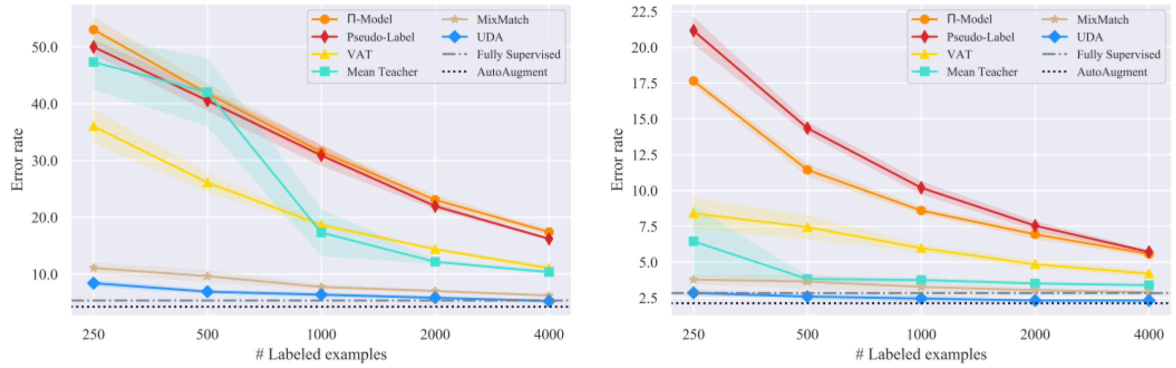
Results with different labeled set sizes



(a) IMDb

(b) Yelp-2

Comparison with semi-supervised learning methods



(a) CIFAR-10 (b) SVHN

Methods	# Sup	Wide-ResNet-28-2	Shake-Shake	ShakeDrop
Supervised	50k	5.4	2.9	2.7
AutoAugment		4.3	2.0	1.5
UDA	4k	5.3	3.6	2.7

- Comparison with semi-supervised learning methods on CIFAR-10 and SVHN with varied number of labeled examples
- UDA performance tested on different architectures

ImageNet Experiments

- Conduct experiments on two settings with different numbers of supervised examples:
  - (a) Use 10% of the supervised data of ImageNet while using all other data as unlabeled data
  - (b) Consider the fully supervised scenario where we keep all images in ImageNet as supervised data and obtain extra unlabeled data from the JFT dataset

ImageNet Results

Methods	top-1 acc	top-5 acc
Supervised	55.09	77.26
Pseudo-Label [36]‡	-	82.41
VAT [44]‡	-	82.78
VAT + EntMin [44]‡	-	83.39
UDA	**68.66**	**88.52**

- UDA improves the top-1 and top-5 accuracy from 55.09% to 68.66% and from 77.26% to 88.52% respectively
- Authors expect that there will be further improvements with more unlabeled data

Conclusion

- Overall data augmentation and semi-supervised learning are well connected: better data augmentation can lead to significantly better semi-supervised learning
- Development of UDA allowing data augmentation with unlabeled data, thus advancing potential for semi-supervised approaches
- TSA that effectively prevents overfitting when much more unlabeled data is available than labeled data
- Significant leaps in performance compared to previous methods in a range of vision and language tasks

Questions?