Unsupervised Data Augmentation For Consistency Training

Liang Shang

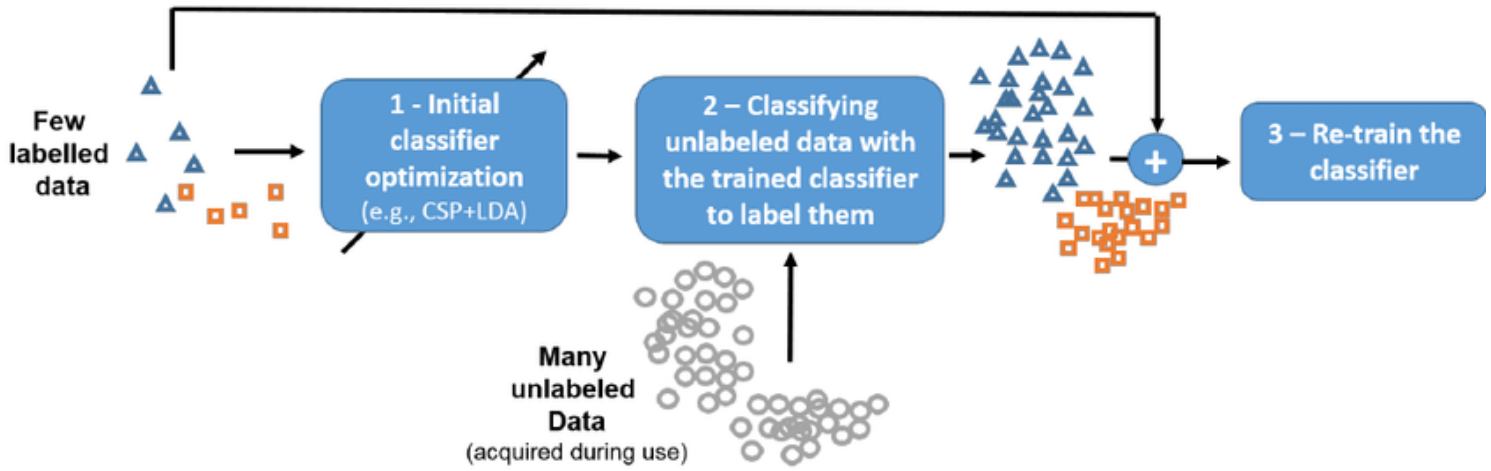
Siyang Chen

Introduction

- We are introducing unsupervised data augmentation (UDA), an augmentation method that focus on the quality of injected noise, which delivers substantial improvements in unsupervised training results.
- UDA substitutes simple noising operation (such as simple Gaussian or dropout noise) with advanced data augmentation methods (such as RandAugment and back-translation).
- UDA performs better on six classification tasks:
  - Text classification: IMDb, Yelp-2, Yelp-5, Amazon-2, Amazon-5
  - Image classification: CIFAR-10, SVHN

Background

- Semi-supervised learning has shown promising improvements in deep learning models when labeled data is scarce.
- Common recent approaches involves using of consistent training on large amount of unlabeled data to constraint model predictions to be invariant to input noise.



Consistency Training

- Consistency training regularizes model predictions to be invariant to small noises to either input examples or hidden states. (This make the model robust to any small changes)
- Most methods under this framework differs in how and where the noise injection is applied.
- Advanced data augmentation methods used in supervised learning also perform well in semi-supervised learning. (Strong correlation present)

Supervised Data Augmentation

- let $q(\hat{x} \mid x)$ be the augmentation transformation from which one can draw augmented examples $\hat{x}$ based on an original example x. It is required that any example $\hat{x} \sim q(\hat{x} \mid x)$ drawn from the distribution shares the same ground-truth label as x.
- Equivalent to constructing an augmented labeled set from the original supervised set and then training the model on the augmented set. (The augmented set needs to provide additional inductive biases to be more effective).
- Despite promising results, data augmentation only provides a steady but limited performance boost because these augmentations has only been applied to a set of small-size labeled examples. This limitation motivated semi-supervised learning where abundant data is available.

Unsupervised Data Augmentation (UDA)

- Given an input x, compute the output distribution $p_\theta(y \mid x)$ given x and a noised version $p_\theta(y \mid x, \epsilon)$ by injecting a small noise ϵ. The noise can be applied to x or hidden states.
- Minimize a divergence metric between the two distributions $\mathcal{D}\left(p_\theta(y \mid x) \parallel p_\theta(y \mid x, \epsilon)\right)$.

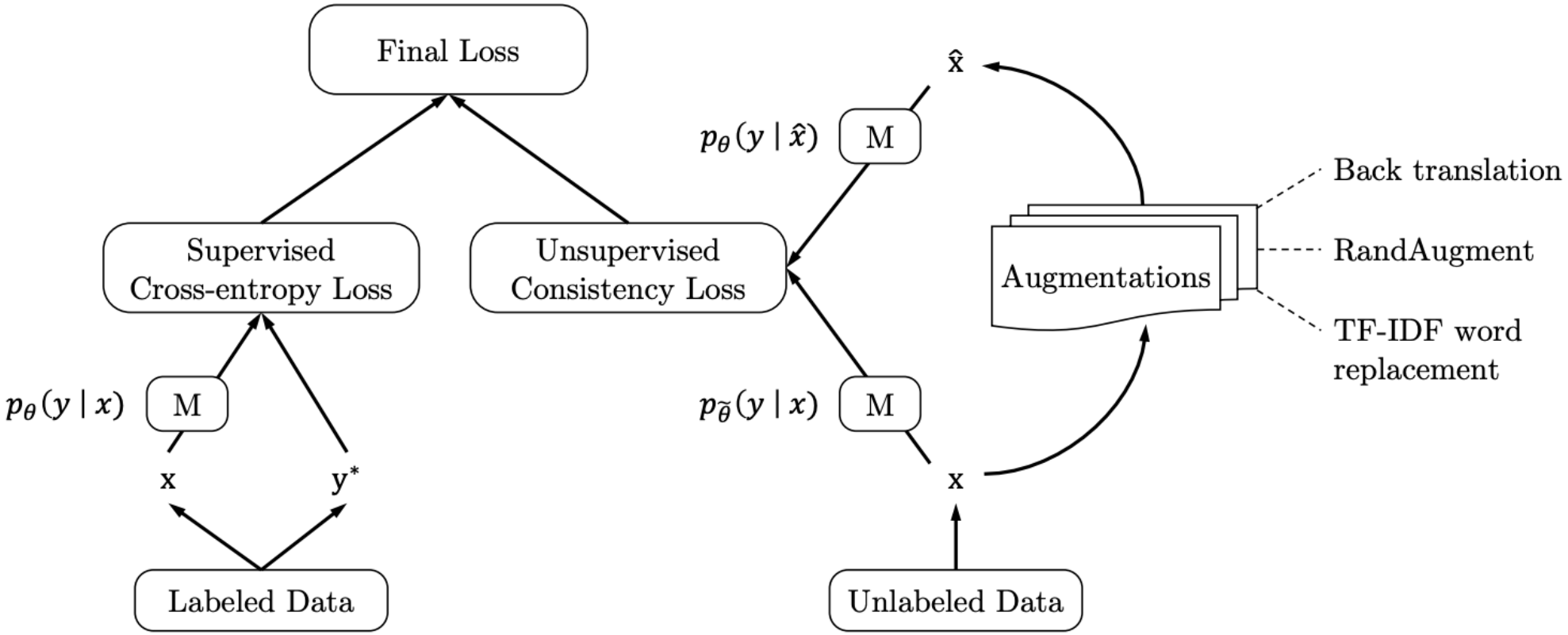
- This procedure enforces the model to be insensitive to the noise. This is essentially minimizing the consistency loss gradually propagates label information from labeled examples to unlabeled ones.
- The UDA presented in this paper focus on the ‘quality’ of the noise operation and its influence on performance of consistency training network. The mechanism is explained in the following slide.

The UDA Mechanism

- Utilize a weighting factor λ when trained with labeled examples. This is used to balance the supervised cross entropy and the unsupervised consistency training loss. As shown in the expression below:

$$\min_{\theta} \; \mathcal{J}(\theta) = \mathbb{E}_{x \sim p_L(x)} \left[-\log p_\theta(f^*(x) \mid x) \right] + \lambda \mathbb{E}_{x \sim p_U(x)} \mathbb{E}_{\hat{x} \sim q(\hat{x}|x)} \left[\mathrm{CE} \left(p_{\tilde{\theta}}(y \mid x) \| p_\theta(y \mid \hat{x}) \right) \right]$$

- CE: cross entropy
- q(x̂ | x): a data augmentation transformation
- $\tilde{\theta}$ is a fixed copy of current parameter θ indicating gradient is not propagated through $\tilde{\theta}$



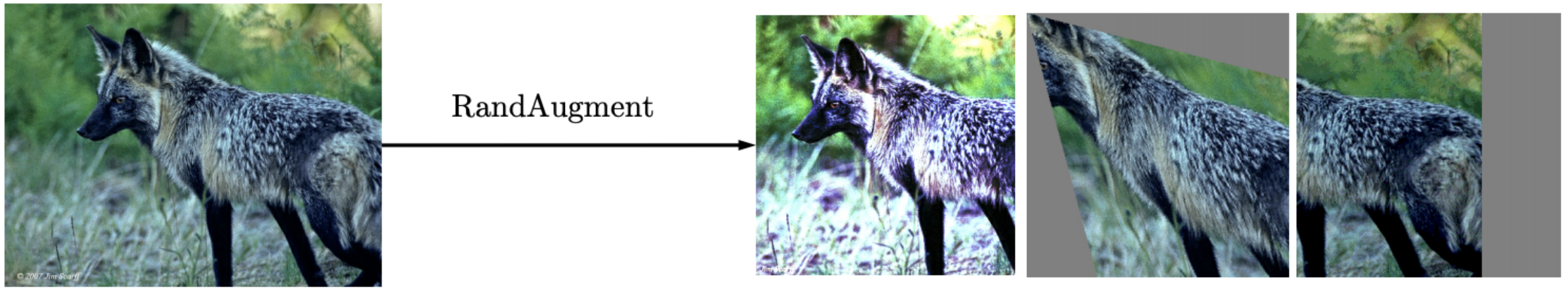
Training objective for UDA, where M is a model that predicts a distribution of y given x.

Advantage of advanced data augmentation

- Valid noise: Advanced data augmentation methods generates realistic augmented examples that share the same ground-truth labels with the original example.
- Diverse noise: Advanced data augmentation can generate a diverse set of examples since it can make large modifications of the input example without changing its label.
- Targeted inductive biases: Data augmentation operations that work well in supervised training essentially provides the missing inductive biases.

Augmentation Strategies – Image Classification

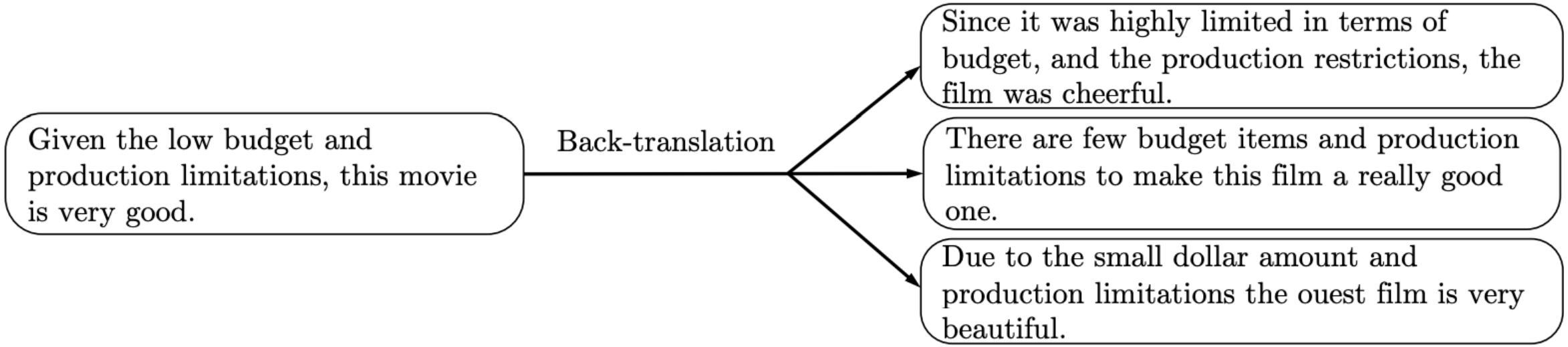
- RandAugment is used for image data augmentation.
- Instead of searching, RandAugment sample uniformly from the Python Image Library (PIL).
- This makes RandAugment simpler and requires no labeled data as there is no need to search for optimal policies.



Augmentation Strategies- Text Classification

- Back-Translation is used for text classification.
- The procedure is translating an existing example x in language A into another language B and then translating it back into A to obtain an augmented example $\hat{x}$.
- Back-translation can generate diverse paraphrases while preserving the semantics of the original sentence, which improves performance.
- A random sampling with a tunable temperature is used for the generation.

Augmentation Strategies- Text Classification



Word replacing with TF-IDF for text classification

- Simple back-translation has little control over which words will be retained, but this requirement is important for topic classification tasks (some key words are more informative than others).
- To address this problem, UDA replaces uniformative words with low TF-IDF scores while keeping those with high TF-IDF values.

Additional Training Techniques- Confidence based masking

- Examples that the current model is not confident about is masked.
- This is done by controlling the calculation of consistency loss in each minibatch.
- Specifically, consistency loss is computed only on examples whose highest probability among classification categories is greater than a threshold β.
- This threshold β is set to a high value to avoid calculating unsure models.

Additional Training Techniques- Sharpening Predictions

- Regularizing predictions to have low entropy is beneficial, thus prediction sharpening is done when computing the target distribution on unlabeled examples by using low temperature τ.
- Loss on unlabeled examples $\mathbb{E}_{x\sim p_U(x)}\mathbb{E}_{\hat{x}\sim q(\hat{x}|x)}\left[\mathrm{CE}\left(p_{\tilde{\theta}}(y \mid x)\|p_\theta(y \mid \hat{x})\right)\right]$ is computed as:

$$\frac{1}{|B|}\sum_{x\in B} I(\max_{y'} p_{\tilde{\theta}}(y' \mid x) > \beta)\mathrm{CE}\left(p_{\tilde{\theta}}^{(sharp)}(y \mid x)\|p_\theta(y \mid \hat{x})\right)$$

$$p_{\tilde{\theta}}^{(sharp)}(y \mid x) = \frac{\exp(z_y/\tau)}{\sum_{y'}\exp(z_{y'}/\tau)}$$

I(·) is the indicator function, Z_y is the logit of label y for example x

Additional Training Techniques- Domain Relevance Data Filtering

- Class distributions of out-of-domain data are mismatched with those of in-domain data, so simply use out-of domain unlabeled data is not sufficient.
- To obtain data relevant to the domain for task at hand, the baseline model trained on the in-domain data is used to infer the labels of data in a large out-of-domain dataset and the examples our model is most confident are picked out.
- This is essentially sorting all examples based on classified probability (for each category) and select the examples with the highest probabilities of being in that category.

Theoretical Analysis - Assumptions

Notations:

- p_U: distribution of unlabeled data
- p_L: distribution of labeled data
- f^*: optimal classifier
- $q(\hat{x}|x)$: augmentation distribution

Theoretical Analysis- Assumptions

- **In-domain** augmentation: data examples generated by data augmentation have non-zero probability under p_U, i.e., $p_U(\hat{x}) > 0$ for $\hat{x} \sim q(\hat{x}|x), x \sim p_U(x)$
- **Label-preserving** augmentation: data augmentation preserves the label of the original example, i.e., $f^*(x) = f^*(\hat{x})$ for $\hat{x} \sim q(\hat{x}|x), x \sim p_U(x)$
- **Reversible** augmentation: the data augmentation operation can be reversed, i.e., $q(\hat{x}|x) > 0 \Leftrightarrow q(x|\hat{x}) > 0$

Theoretical Analysis- Intuition

- For a graph G_{p_U}, where each node corresponds to a data sample $x \in X$ and an edge $(\hat{x}, x)$ exists iff $q(\hat{x}|x) > 0$
- For an N-category classification problem, by an ideal data augmentation method, G_{p_U} should have N components
- For each component C_i of the graph, as long as we have one labeled data, by traversing C_i via augmentation operation $q(\hat{x}|x)$, we can propagate the label over all data in C_i

Theoretical Analysis- Intuition

- In order to find a perfect classifier via such label propagation, there should exist at least one labeled example in each component.
- Which means the number of components is the lower bound the minimum amount of labeled examples needed.
- With a better augmentation method, the number of components can be decreased.

Theoretical Analysis - Theorem

P_i: the total probability that a labeled data point fall into the i-th components, i.e., $P_i = \sum_{x \in C_i} P_L(x)$

Theorem 1. *Under UDA, let $Pr(\mathcal{A})$ denote the probability that the algorithm cannot infer the label of a new test example given m labeled examples from P_L. $Pr(\mathcal{A})$ is given by*

$$Pr(\mathcal{A}) = \sum_i P_i (1 - P_i)^m.$$

In addition, $O(k/\epsilon)$ labeled examples can guarantee an error rate of $O(\epsilon)$, i.e.,

$$m = O(k/\epsilon) \implies Pr(\mathcal{A}) = O(\epsilon).$$

Experiments

Step 1: Correlation between supervised and semi-supervised performance

Augmentation (# Sup examples)	Sup (50k)	Semi-Sup (4k)
Crop & flip	5.36	10.94
Cutout	4.42	5.43
RandAugment	**4.23**	**4.32**

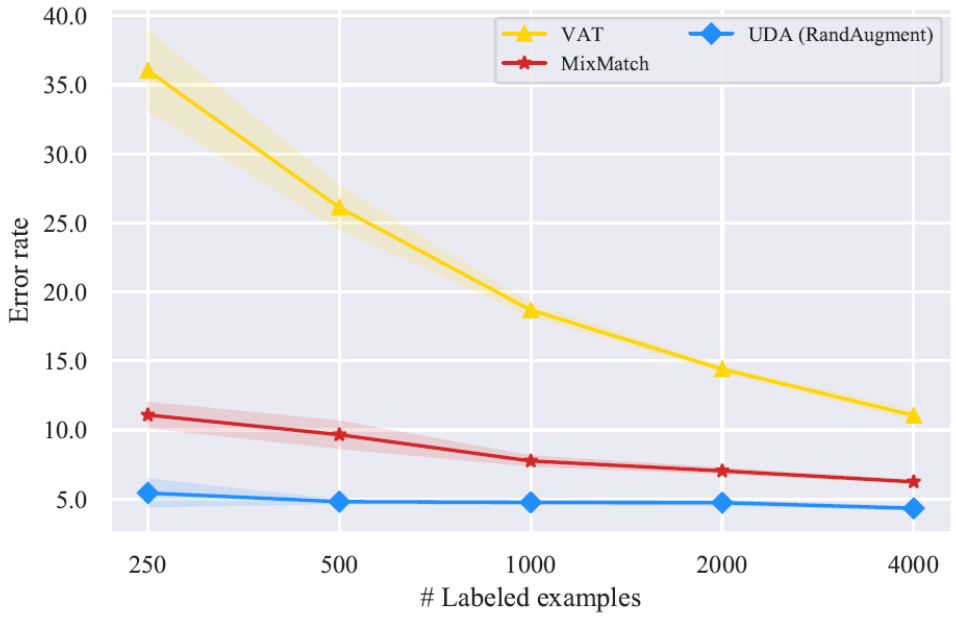
Table 1: Error rates on CIFAR-10.

Augmentation (# Sup examples)	Sup (650k)	Semi-sup (2.5k)
✗	38.36	50.80
Switchout	37.24	43.38
Back-translation	**36.71**	**41.35**

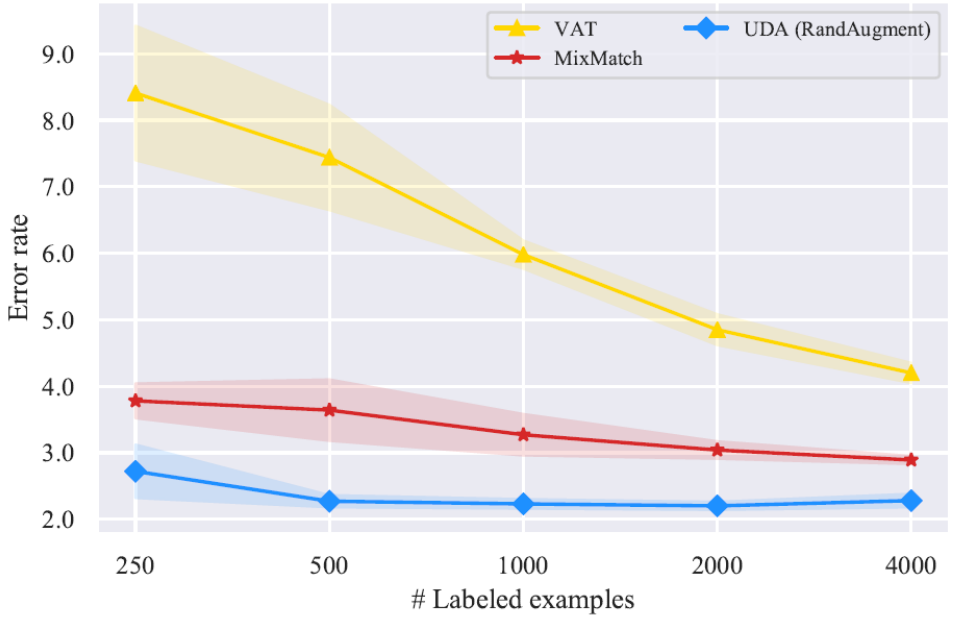
Table 2: Error rate on Yelp-5.

Experiments

Step 2: Vision semi-supervised learning benchmarks – Vary the size



(a) CIFAR-10



(b) SVHN

Experiments

Step 2: Vision semi-supervised learning benchmarks – Vary the model

Method	**Model**	**# Param**	CIFAR-10 (4k)	SVHN (1k)
Π-Model (Laine & Aila, 2016)	Conv-Large	3.1M	12.36 ± 0.31	4.82 ± 0.17
Mean Teacher (Tarvainen & Valpola, 2017)	Conv-Large	3.1M	12.31 ± 0.28	3.95 ± 0.19
VAT + EntMin (Miyato et al., 2018)	Conv-Large	3.1M	10.55 ± 0.05	3.86 ± 0.11
SNTG (Luo et al., 2018)	Conv-Large	3.1M	10.93 ± 0.14	3.86 ± 0.27
VAdD (Park et al., 2018)	Conv-Large	3.1M	11.32 ± 0.11	4.16 ± 0.08
Fast-SWA (Athiwaratkun et al., 2018)	Conv-Large	3.1M	9.05	-
ICT (Verma et al., 2019)	Conv-Large	3.1M	7.29 ± 0.02	3.89 ± 0.04
Pseudo-Label (Lee, 2013)	WRN-28-2	1.5M	16.21 ± 0.11	7.62 ± 0.29
LGA + VAT (Jackson & Schulman, 2019)	WRN-28-2	1.5M	12.06 ± 0.19	6.58 ± 0.36
mixmixup (Hataya & Nakayama, 2019)	WRN-28-2	1.5M	10	-
ICT (Verma et al., 2019)	WRN-28-2	1.5M	7.66 ± 0.17	3.53 ± 0.07
MixMatch (Berthelot et al., 2019)	WRN-28-2	1.5M	6.24 ± 0.06	2.89 ± 0.06
Mean Teacher (Tarvainen & Valpola, 2017)	Shake-Shake	26M	6.28 ± 0.15	-
Fast-SWA (Athiwaratkun et al., 2018)	Shake-Shake	26M	5.0	-
MixMatch (Berthelot et al., 2019)	WRN	26M	4.95 ± 0.08	-
UDA (RandAugment)	WRN-28-2	1.5M	4.32 ± 0.08	$\mathbf{2.23 \pm 0.07}$
UDA (RandAugment)	Shake-Shake	26M	3.7	-
UDA (RandAugment)	PyramidNet	26M	**2.7**	-

Experiments

Step 3: Text semi-supervised classification tasks

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
$\text{BERT}_{\text{BASE}}$	✗	18.40	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**	-

Experiments

Step 4: Scalability test on the ImageNet dataset

I. Use 10% of the supervised data of ImageNet while using all other data as unlabeled data.

II. Use all images in ImageNet as supervised data, use data filtered by domain-relevance data filtering method as unlabeled data.

Methods	SSL	10%	100%
ResNet-50	✗	55.09 / 77.26	77.28 / 93.73
w. RandAugment		58.84 / 80.56	78.43 / 94.37
UDA (RandAugment)	✓	**68.78 / 88.80**	**79.05 / 94.49**

Conclusion

- Data augmentation and semi-supervised learning are well connected, better data augmentation can lead to significantly better semi-supervised learning.
- UDA generate diverse and realistic noise and enforces the model to be consistent with respect to these noises.
- UDA combines well with representation learning and nearly matches the performance of fully supervised models trained on full labeled sets which are larger by one magnitude in size.