Unsupervised Data Augmentation for Consistency Training

Носов Степан, группа 162

Москва, 2019

Consistency Training

- Хотим повысить устойчивость нашей модели к шуму(увеличить margin)
- Для этого совершим преобразование $\hat{x} = q(x, \epsilon)$
- Будем минимизировать $D = L_{sup}(x, y) + L_{con}(x, \hat{x})$

Inductive bias

- Inductive bias - это набор предположений о данных, позволяющий модели иметь обобщающую способность.

- Когда мы совершаем аугментацию данных - мы закладываем новые inductive bias в модель

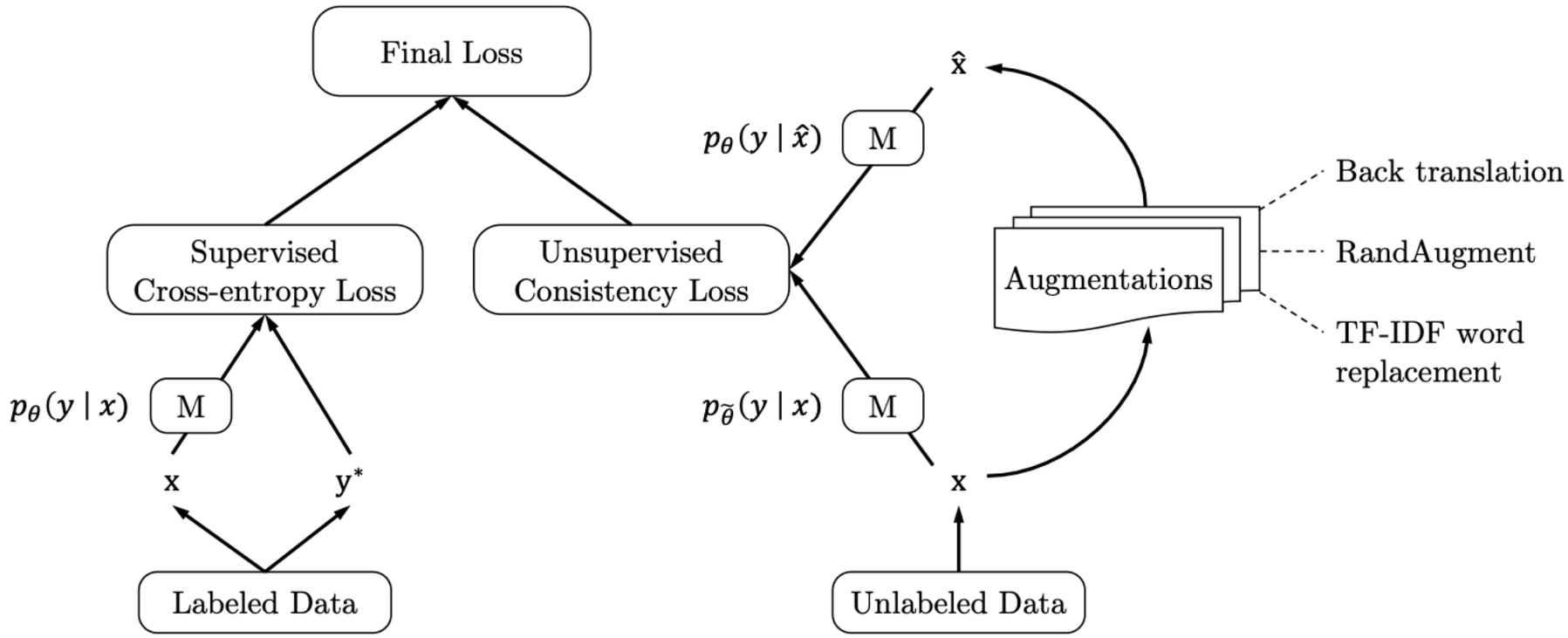
- Аугментация дает ограниченный прирост в supervised-обучении, так как применяет на очень ограниченном наборе данных

Unsupervised Data Augmentation(UDA)

- Давайте использовать механизмы аугментации данных для consistency training

- Теория - аугментации, хорошо работавшие для supervised задачи, будут хорошо работать и для unsupervied

UDA


Final Loss
Supervised
Cross-entropy Loss
Unsupervised
Consistency Loss
$p_\theta(y \mid x)$
M
x
y^*
Labeled Data
$\hat{x}$
$p_\theta(y \mid \hat{x})$
M
$p_{\tilde{\theta}}(y \mid x)$
M
x
Unlabeled Data
Augmentations
Back translation
RandAugment
TF-IDF word
replacement

UDA

$$\min_{\theta} \; \mathcal{J}(\theta) = \mathbb{E}_{x, y^* \in L} \left[-\log p_\theta(y^* \mid x) \right] + \lambda \mathbb{E}_{x \in U} \mathbb{E}_{\hat{x} \sim q(\hat{x} \mid x)} \left[\mathcal{D}_{\mathrm{KL}} \left(p_{\tilde{\theta}}(y \mid x) \;\middle\|\; p_\theta(y \mid \hat{x}) \right) \right].$$

- Здесь $\tilde{\theta}$ - это фиксированная копия параметров, которая игнорируется при вычислении градиента, λ - параметр для балансировки лоссов.

UDA

Несколько интуиций с использованием продвинутых аугментаций:

- Продвинутые методы генерируют правдоподобные объекты, которые разделяют с исходным класс, так что это безопасно - делать сильный акцент на consistency loss

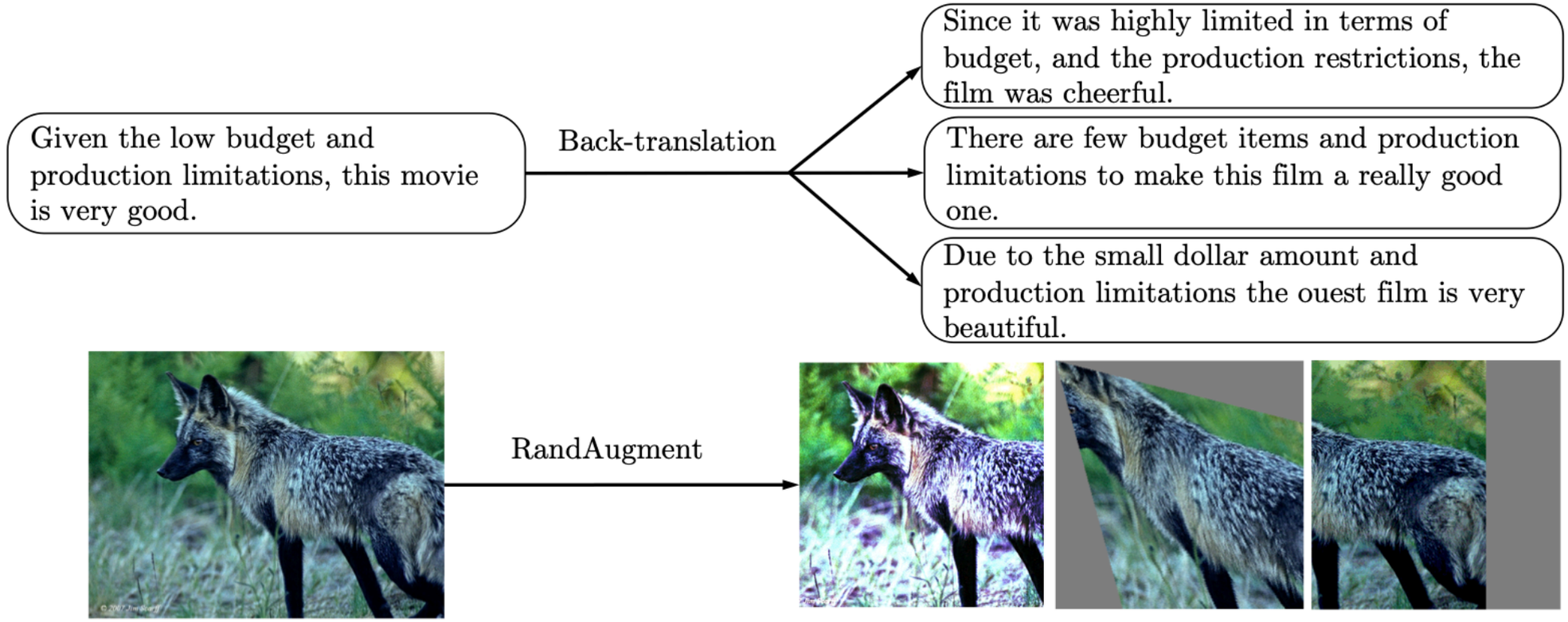
- Продвинутые методы генерируют более разнообразные объекты, чем, например, тривиальное гауссовское зашумление, так что вычисление на них consistency loss обеспечивает серьезное улучшение

- Продвинутые аугментации привносят в себе новые inductive bias, которые позволяют повысить обобщающую способность модели

Виды аугментаций

- *RandAugment for Image Classification* - равномерное семплирование из набора трансформаций из PIL

- *Back-translation for Text Classification* - переводим текст на другой язык, а затем переводим обратно



Виды аугментаций

- *Word Replacing with TF-IDF for Text Classification* - заменяем слова с низким TF-IFD, на слова с высоким TF-IDF

Training Signal Annealing

- Модель быстро переобучается на размеченных данных и недообучается на неразмеченных

- Если $p_\theta(y^* \mid x) > \eta_t$, то выкидываем объект из loss'а.

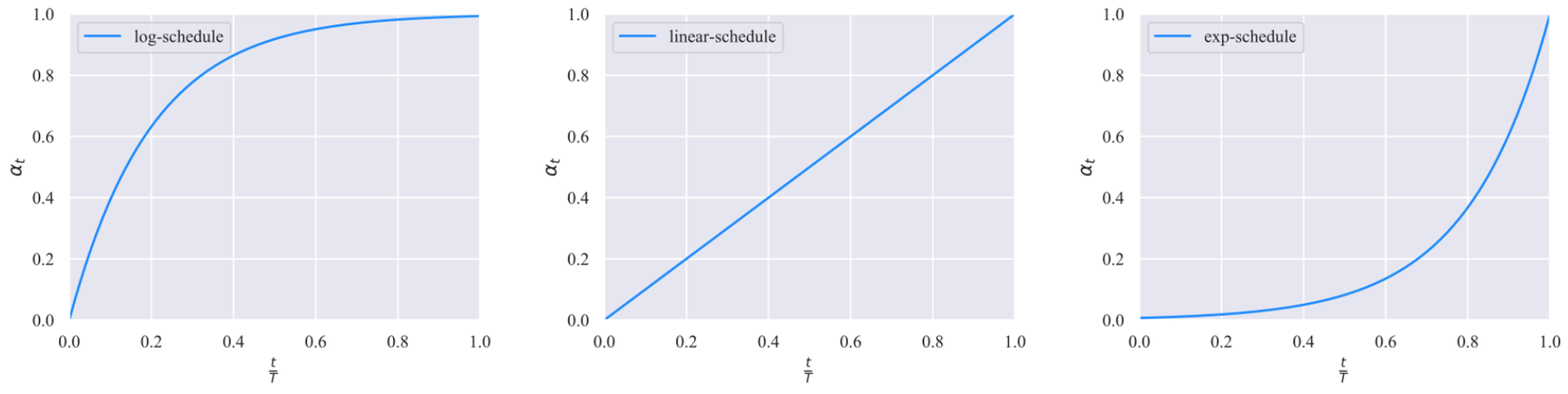


Figure 3: Three schedules of TSA. We set $\eta_t = \alpha_t * (1 - \frac{1}{K}) + \frac{1}{K}$. α_t is set to $1 - \exp(-\frac{t}{T} * 5)$, $\frac{t}{T}$ and $\exp((\frac{t}{T} - 1) * 5)$ for the log, linear and exp schedules.

Эксперименты

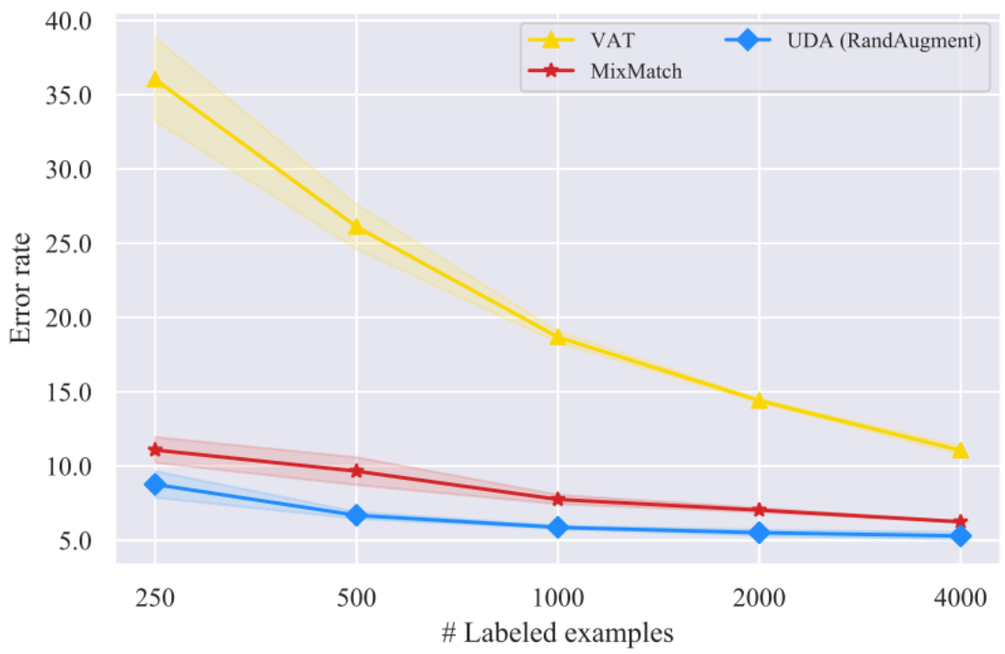
Augmentation (# Sup examples)	Sup (50k)	Semi-Sup (4k)
Crop & flip	5.36	16.17
Cutout	4.42	6.42
RandAugment	**4.23**	**5.29**

Table 1: Error rates on CIFAR-10.

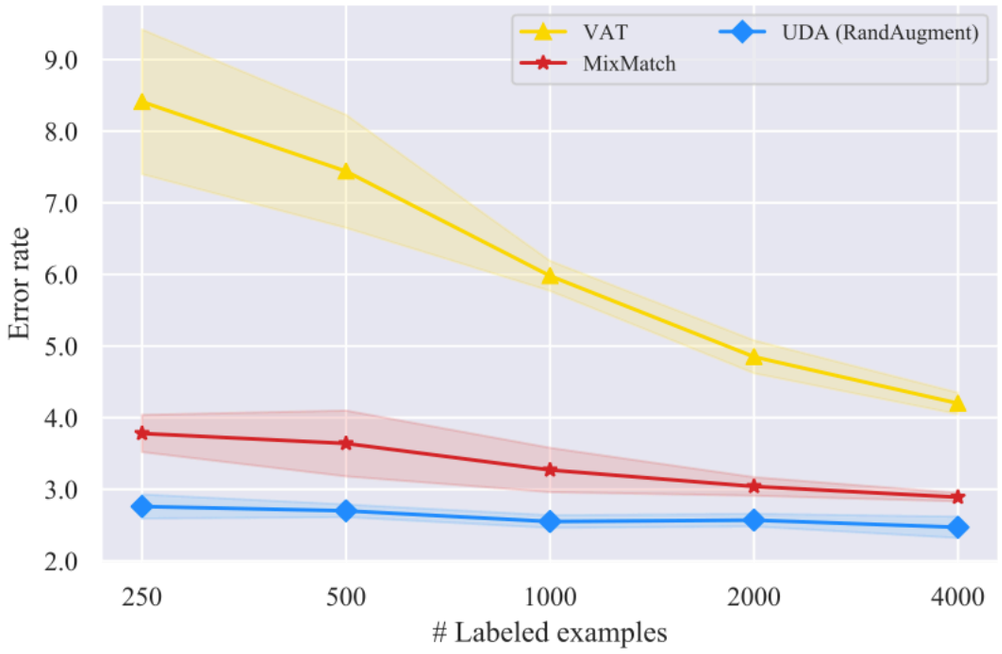
Augmentation (# Sup examples)	Sup (650k)	Semi-sup (2.5k)
✗	38.36	50.80
Switchout	37.24	43.38
Back-translation	**36.71**	**41.35**

Table 2: Error rate on Yelp-5.

Эксперименты



(a) CIFAR-10



(b) SVHN

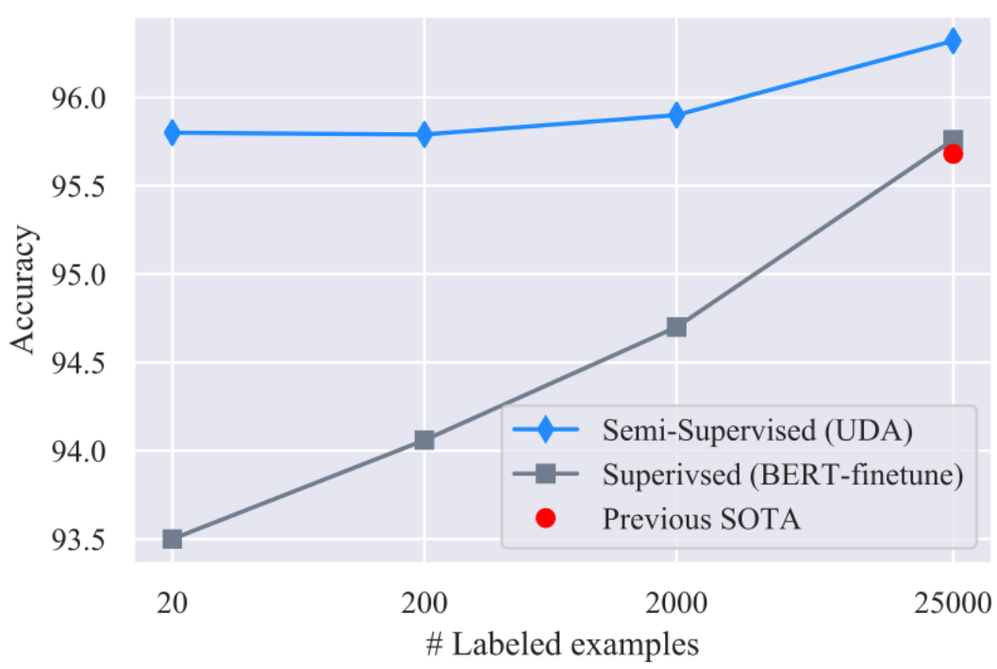
Method	**Model**	**# Param**	CIFAR-10 (4k)	SVHN (1k)
Π-Model (Laine & Aila, 2016)	Conv-Large	3.1M	12.36 ± 0.31	4.82 ± 0.17
Mean Teacher (Tarvainen & Valpola, 2017)	Conv-Large	3.1M	12.31 ± 0.28	3.95 ± 0.19
VAT + EntMin (Miyato et al., 2018)	Conv-Large	3.1M	10.55 ± 0.05	3.86 ± 0.11
SNTG (Luo et al., 2018)	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	5.29 ± 0.25	**2.55 ± 0.09**
UDA (RandAugment)	Shake-Shake	26M	3.7	-
UDA (RandAugment)	PyramidNet	26M	**2.7**	-

Table 3: Comparison between methods using different models where PyramidNet is used with ShakeDrop regularization. On CIFAR-10, with only 4,000 labeled examples, UDA matches the performance of fully supervised Wide-ResNet-28-2 and PyramidNet+ShakeDrop, where they have an error rate of 5.4 and 2.7 respectively when trained on 50,000 examples without RandAugment. On SVHN, UDA also matches the performance of our fully supervised model trained on 73,257 examples without RandAugment, which has an error rate of 2.84.

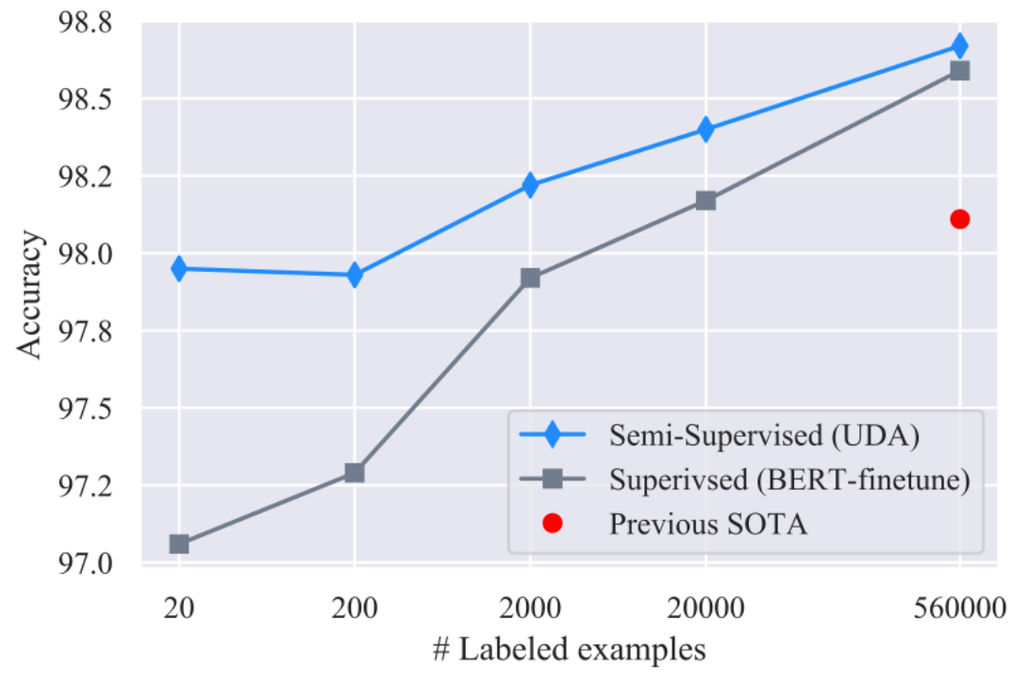
Эксперименты

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**	-



(a) IMDb



(b) Yelp-2

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**

TSA efficiency

TSA schedule	Yelp-5	CIFAR-10
✗	50.81	5.67
log-schedule	49.06	5.67
linear-schedule	45.41	**5.29**
exp-schedule	**41.35**	7.81

Выводы

- Чем лучше аугментация, тем лучше ее эффект от применения в SSL
- UDA показывает хорошие результаты в transfer learning
- UDA способен показывать высокое качество на минимальном числе размеченных данных

Вопросы

- Запишите и поясните целевую функцию, используемую в статье UNSUPERVISED DATA AUGMENTATION FOR CONSISTENCY TRAINING
- Кратко опишите как минимум два вида аугментаций, используемых в UDA
- Опишите технику Training Signal Annealing