Uncertainty-aware Self-training for Few-shot Text Classification

Subhabrata Mukherjee, Ahmed Hassan Awadallah

NeurIPS 2020

14th October, 2020

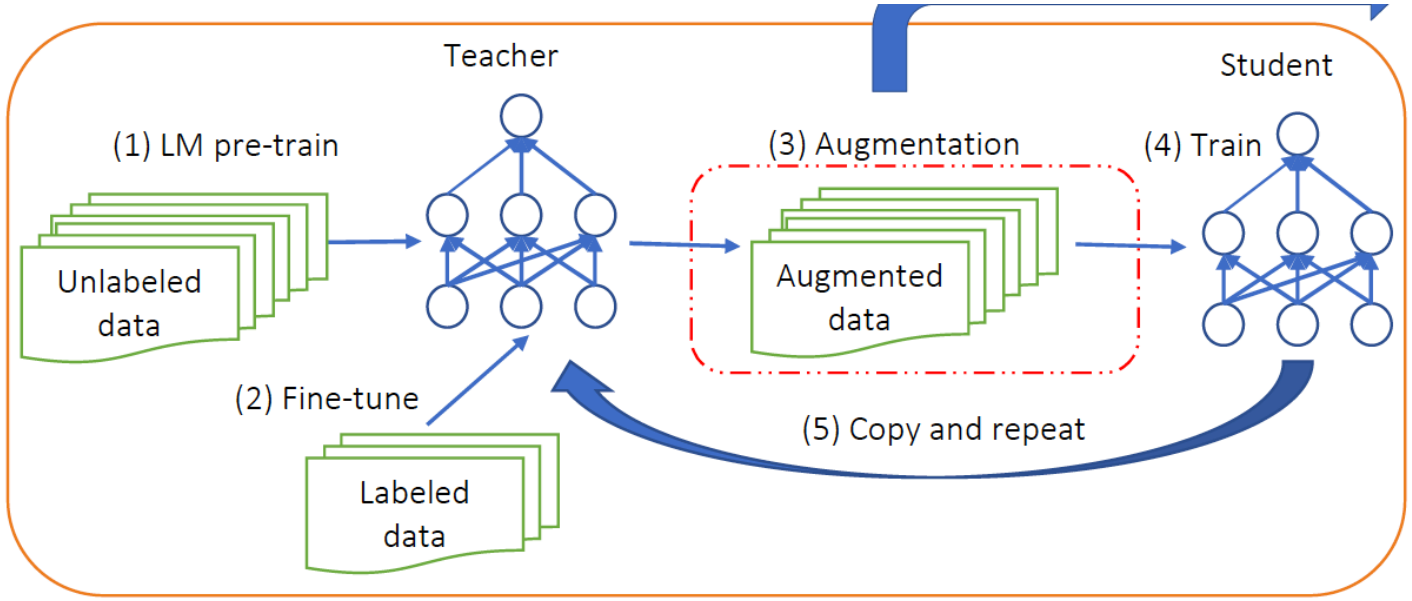
Summary

- Problem: Fine-tuning requires 1000s of labeled samples
- Baseline: Self-training, Backtranslation
- Cons: Self-training focuses mostly on easy samples (little to gain)
  - Ignores hard samples (lot to gain)
- Challenge: Hard samples could be noisy or too difficult to learn from
  - Not trivial to generate uncertainty estimates for non-probabilistic models
- Proposal: Leverage bayesian deep learning to obtain uncertainty estimates of the teacher for pseudo-labeling
- Result: 20-30 labeled samples per class approx 3% of fine-tuning 1000s

Background

- $D_l = \{x_i, y_i\}$
- n labeled instances
- y_i being the class label for x_i
- $x_i = \{x_{i1}, \ldots x_{im}\}$, sequence of m tokens
- $D_u = \{x_j\}$, N unlabeled instances
- $n << N$

Self-training



- Subset $S_u \subset D_u$
- Subset selection based on confidence scores of the teacher model

$$min_W \mathbb{E}_{x_l, y_l \in D_l}[-log\ p(y_l|x_l; W)] + \lambda \mathbb{E}_{x_u \in S_u, S_u \subset D_u} \mathbb{E}_{y \sim p(y|x_u; W^*)}[-log\ p(y|x_u; W)]$$

Bayesian Neural Network

- Weights are random variables
  - Prior distribution over the parameters
- Neural net inference

$$P(y = c|x, W) = softmax(f^W(x))$$

- Posterior distribution, p(W|X,Y)
- Bayesian inference

$$p(y = c|x) = \int_W p(y = c|f^W(x))p(W|X, Y)dW$$

- Surrogate distribution, $q_\theta(W)$ approx. by dropout dist. $\{\widetilde{W}_t\}_{t=1}^{T} \sim q_\theta(W)$

$$p(y = c|x) \approx p(y = c|f^W(x))q_\theta(W)dW$$

$$\approx \frac{1}{T}\sum_{t=1}^{T} p(y = c|f^{\widetilde{W}_t}(x)) = \frac{1}{T}\sum_{t=1}^{T} softmax(f^{\widetilde{W}_t}(x))$$

T	p(y=1)	p(y=2)	p(y=3)
1	0.33	0.33	0.33
2	0.5	0.2	0.3
3	0.98	0.01	0.01

Uncertainty-aware self-training

T	p(y=1)	p(y=2)	p(y=3)
1	0.33	0.33	0.33
2	0.5	0.2	0.3
3	0.98	0.01	0.01
E(y)	0.6	0.18	0.21

- Surrogate distribution, $q_\theta(W)$ approx. by dropout dist

$$p(y=c|x) \approx p(y=c|f^W(x))q_\theta(W)dW$$

$$\approx \frac{1}{T}\sum_{t=1}^{T} p(y=c|f^{\widetilde{W}_t}(x)) = \frac{1}{T}\sum_{t=1}^{T} softmax(f^{\widetilde{W}_t}(x))$$

- Compute uncertainty score of an unlabeled instance

$$p(y_t{}^*) = softmax(f^{\widetilde{W}_t}(x_u)) \qquad E(y) = \frac{1}{T}\sum_{t=1}^{T} softmax(f^{\widetilde{W}_t}(x))$$

$$y_u = argmax_c \sum_{t=1}^{T} \mathbb{I}[argmax_{c'}(p(y_t{}^* = c')) = c]$$

$$min_W \mathbb{E}_{x_l,y_l\in D_l}[-log\, p(y_l|x_l;W)] + \lambda \mathbb{E}_{x_u\in S_u, S_u\subset D_u} \mathbb{E}_{y\sim p(y|x_u;W^*)}[-log\, p(y|x_u;W)]$$

$$min_{W,\theta} \mathbb{E}_{x_u\in S_u, S_u\subset D_u} \mathbb{E}_{\widetilde{W}\sim q_\theta(W^*)} \mathbb{E}_{y\sim p(y|f^{\widetilde{W}}(x_u))}[-log\, p(y|f^W(x_u)]$$

Sample selection

T	p(y=1)	p(y=2)	p(y=3)
1	0.97	0.02	0.01
2	0.96	0.02	0.02
3	0.98	0.01	0.01
E(y)	0.97	0.02	0.01

T	p(y=1)	p(y=2)	p(y=3)
1	0.33	0.33	0.33
2	0.33	0.33	0.33
3	0.33	0.33	0.33
E(y)	0.33	0.33	0.33

- Based on Bayesian Active Learning by Disagreement (BALD)
- Select samples that maximize info. gain about model parameters
  - Max. infogain between predictions and model posterior

$$\mathbb{B}(y_u, W|x_u, D'_u) = \mathbb{H}[y_u|x_u, D'_u] - \mathbb{E}_{p(W|D'_u)}[\mathbb{H}[y_u|x_u, W]]$$

$$\widehat{\mathbb{B}}(y_u, W|x_u, D'_u) = -\sum_c (\frac{1}{T}\sum_t \widehat{p}_c^t) log(\frac{1}{T}\sum_t \widehat{p}_c^t) + \frac{1}{T}\sum_{t,c} \widehat{p}_c^t log(\widehat{p}_c^t)$$

where, $\widehat{p}_c^t = p(y_u = c|f^{\widetilde{W}_t}(x_u) = softmax(f^{\widetilde{W}_t}(x_u))$.

  - High value indicates that teacher model is highly confused about the expected label of the instance

Sample selection

- Class-dependent selection
  - Avoids sampling instances from harder class
  - Create pseudo-labeled set for every class
- Selection with exploration
  - Without replacement with probability

$$p_{u,c}^{easy} = \frac{1 - \widehat{\mathbb{B}}(y_u, W | x_u, D'_u)}{\sum_{x_u \in S_{u,c}} 1 - \widehat{\mathbb{B}}(y_u, W | x_u, D'_u)} \qquad p_{u,c}^{hard} = \frac{\widehat{\mathbb{B}}(y_u, W | x_u, D'_u)}{\sum_{x_u \in S_{u,c}} \widehat{\mathbb{B}}(y_u, W | x_u, D'_u)}$$

Confident learning

- Prev. method select samples using predictive mean, while ignoring the uncertainty of the model in terms of predictive variance
- Prediction uncertainty of the teacher model is given by the variance of the marginal distribution

$$
\begin{aligned}
Var(y) &= Var[\mathbb{E}(y|W,x)] + \mathbb{E}[Var(y|W,x)] \\
&= Var(softmax(f^{W}(x)) + \sigma^2 \\
&\approx \left(\frac{1}{T}\sum_{t=1}^{T} {y_t}^*(x)^T {y_t}^*(x) - E(y)^T E(y)\right) + \sigma^2
\end{aligned}
$$

$$
{y_t}^*(x) = softmax(f^{\widetilde{W}_t}(x))
$$

$$
E(y) = \frac{1}{T}\sum_{t=1}^{T} {y_t}^*(x).
$$

- Final self-training loss

$$
min_{W,\theta}\, \mathbb{E}_{x_u \in S_u, S_u \subset D_u}\, \mathbb{E}_{\widetilde{W} \sim q_\theta(W^*)}\, \mathbb{E}_{y \sim p(y|f^{\widetilde{W}}(x_u))} [log\, p(y|f^{W}(x_u) \cdot log\, Var(y)]
$$

Confident learnin

T	p(y=1)	p(y=2)	p(y=3)	y_t(x)*y_t(x)
1	0.97	0.02	0.01	[0.97 0.02 0.01] [0.97 0.02 0.01] = 0.9414
2	0.96	0.02	0.02	[0.96 0.02 0.02] [0.96 0.02 0.02] = 0.9224
3	0.98	0.01	0.01	[0.98 0.01 0.01] [0.98 0.01 0.01] = 0.9606
E(y)	0.97	0.02	0.01	0.9415

- Prev. method select samp
  the uncertainty of the mc
- Prediction uncertainty of
  of the marginal distribution

$$Var(y) = Var[\mathbb{E}(y|W,x)] + \mathbb{E}[Var(y|W,x)]$$
$$= Var(softmax(f^W(x)) + \sigma^2$$
$$\approx \left(\frac{1}{T}\sum_{t=1}^{T} {y_t}^*(x)^T {y_t}^*(x) - E(y)^T E(y)\right) + \sigma^2$$

$${y_t}^*(x) = softmax(f^{\widetilde{W}_t}(x))$$

$$E(y) = \frac{1}{T}\sum_{t=1}^{T} {y_t}^*(x).$$

- Final self-training loss

$$min_{W,\theta}\ \mathbb{E}_{x_u \in S_u, S_u \subset D_u}\ \mathbb{E}_{\widetilde{W} \sim q_\theta(W^*)}\ \mathbb{E}_{y \sim p(y|f^{\widetilde{W}}(x_u))}[log\ p(y|f^W(x_u) \cdot log\ Var(y)]$$

Algorithm Summary

Algorithm 1: Uncertainty-aware self-training pseudo-code.

Continue pre-training teacher language model on task-specific unlabeled data D_u ;
Fine-tune model f^W with parameters W on task-specific small labeled data D_l ;
while *not converged* **do**
  Randomly sample S_u unlabeled examples from D_u ;
  for $x \in S_u$ **do**
    for $t \leftarrow 1$ **to** T **do**
      $W_t \sim Dropout(W)$;
      $y_t^* = softmax(f^{W_t}(x))$;
    end
    Compute predictive sample mean $E(y)$ and predictive sample variance $Var(y)$ with Equation 11 ;
    Compute BALD acquisition function with Equation 6 ;
  end
  Sample R instances from S_u employing sample selection with Equations 7 or 8 ;
  Pseudo-label R sampled instances with model f^W ;
  Re-train model on R pseudo-labeled instances with Equation 12 and update parameters W ;
end

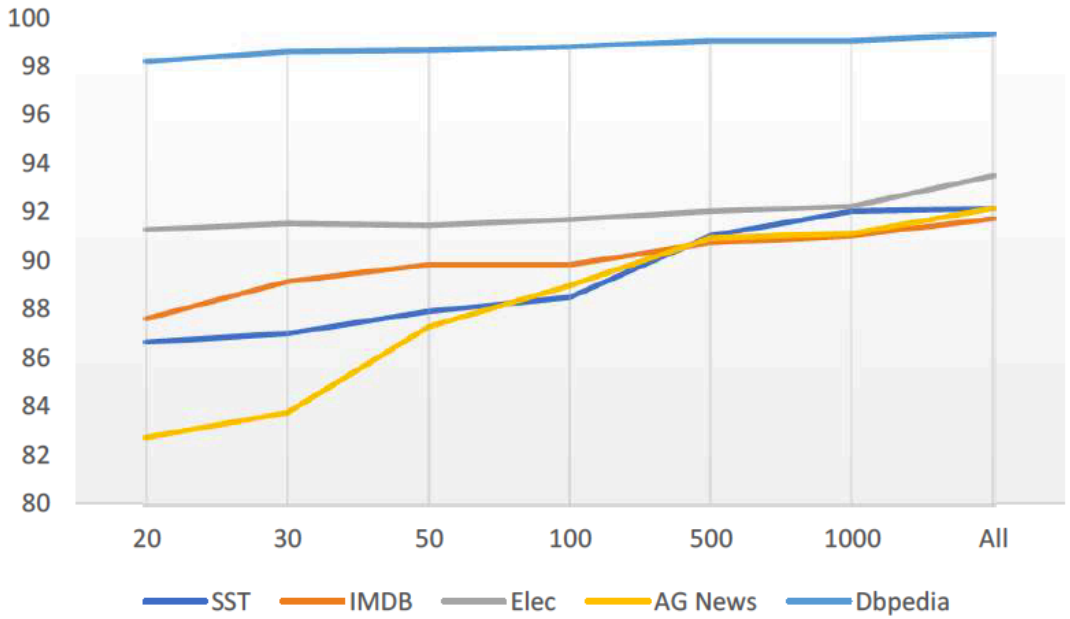
Experiments

- Base model – BERT Train on K=30 instances per class

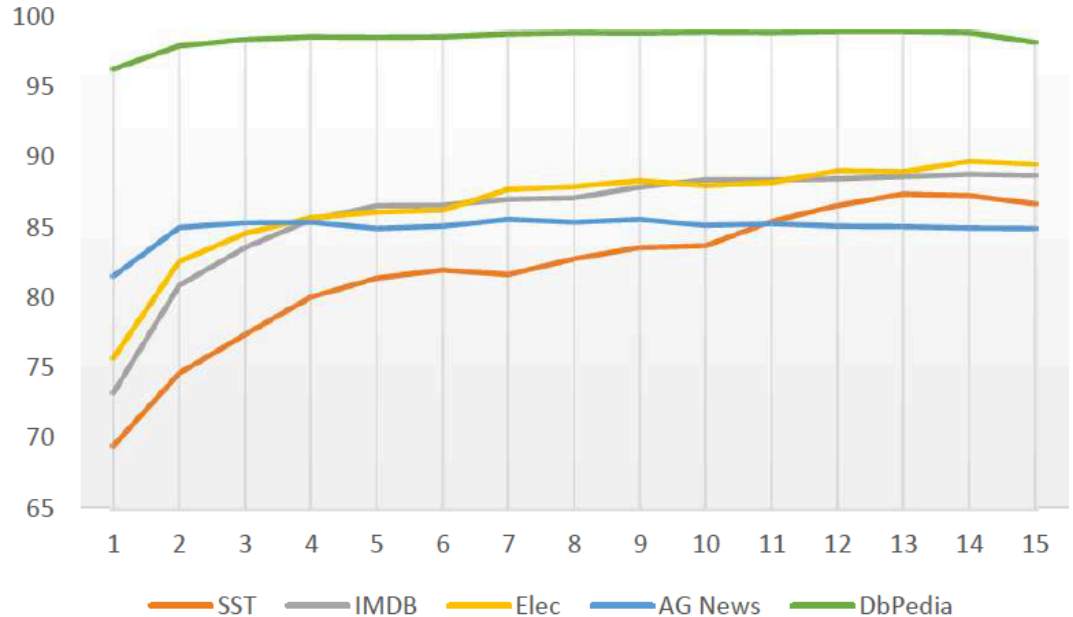
Dataset	All train	30 labels per class for training and for validation			
	BERT	BERT (base)	UDA	Classic ST	UST (our method)
SST	92.12	69.79 (6.45)	83.58 (2.64) (↑ 19.8)	84.81 (1.99) (↑ 21.5)	**88.19** (1.01) (↑ 26.4)
IMDB	91.70	73.03 (6.94)	**89.30** (2.05) (↑ 22.3)	78.97 (8.52) (↑ 8.1)	89.21 (0.83) (↑ 22.2)
Elec	93.46	82.92 (3.34)	89.64 (2.13) (↑ 8.1)	89.92 (0.36) (↑ 8.4)	**91.27** (0.31) (↑ 10.1)
AG News	92.12	80.74 (3.65)	85.92 (0.71) (↑ 6.4)	84.62 (4.81) (↑ 4.8)	**87.74** (0.54) (↑ 8.7)
DbPedia	99.26	97.77 (0.40)	96.88 (0.58) (↓ 0.9)	98.39 (0.64) (↑ 0.6)	**98.57** (0.18) (↑ 0.8)
Average	93.73	80.85 (4.16)	89.06 (1.62) (↑ 10.2)	87.34 (3.26) (↑ 8.0)	**91.00** (0.57) (↑ 12.6)

Ablation Analysis

	SST	IMDB	Elec	AG News	Dbpedia	Average
BERT Base	69.79	73.03	82.92	80.74	97.77	80.85
Classic ST (Uniform)	84.81	78.97	89.92	84.62	98.39	87.34
UST (Easy)	88.19	89.21	**91.27**	**87.74**	**98.57**	**91.00**
- removing *Class*	87.33	87.22	89.18	86.88	98.27	89.78
- removing *Conf*	86.73	**90.00**	90.40	84.17	98.49	89.96
UST (Hard)	88.02	88.49	90.00	85.02	98.56	90.02
- removing *Class*	80.45	89.28	90.07	83.07	98.46	88.27
- removing *Conf*	**88.48**	87.93	88.74	84.45	98.26	89.57



(a) UST accuracy with K train labels/class.



(b) UST accuracy over iterations.

Figure 2: Improvement in UST accuracy with more training labels and epochs.