Calibration of Pre-trained Transformers

Shrey Desai， Greg Durrett
Department of Computer Science The University of Texas at Austin

EMNLP2020

蔡丽

Outline

- Motivation
- Definitions
- Experiments
- Conclusion

Motivation

- Are pre-trained transformers calibration？
- How to calibrate the pre-trained transformers？

Definitions

- A model is **calibrated** if the confidence estimates of its predictions are aligned with empirical likelihoods.
- **perfect calibration** is achieved when
- $$\mathrm{P}(Y = \mathrm{y}|\mathrm{Q} = \mathrm{q}) = \mathrm{q}$$

Confidentc $Q \in R$, labels Y $\in \mathcal{Y}$

- Approximated by **ECE**[1](expected calibration error)

[1] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In Proceedings of the International Conference on Machine Learning (ICML).

On Calibration of Modern Neural Networks

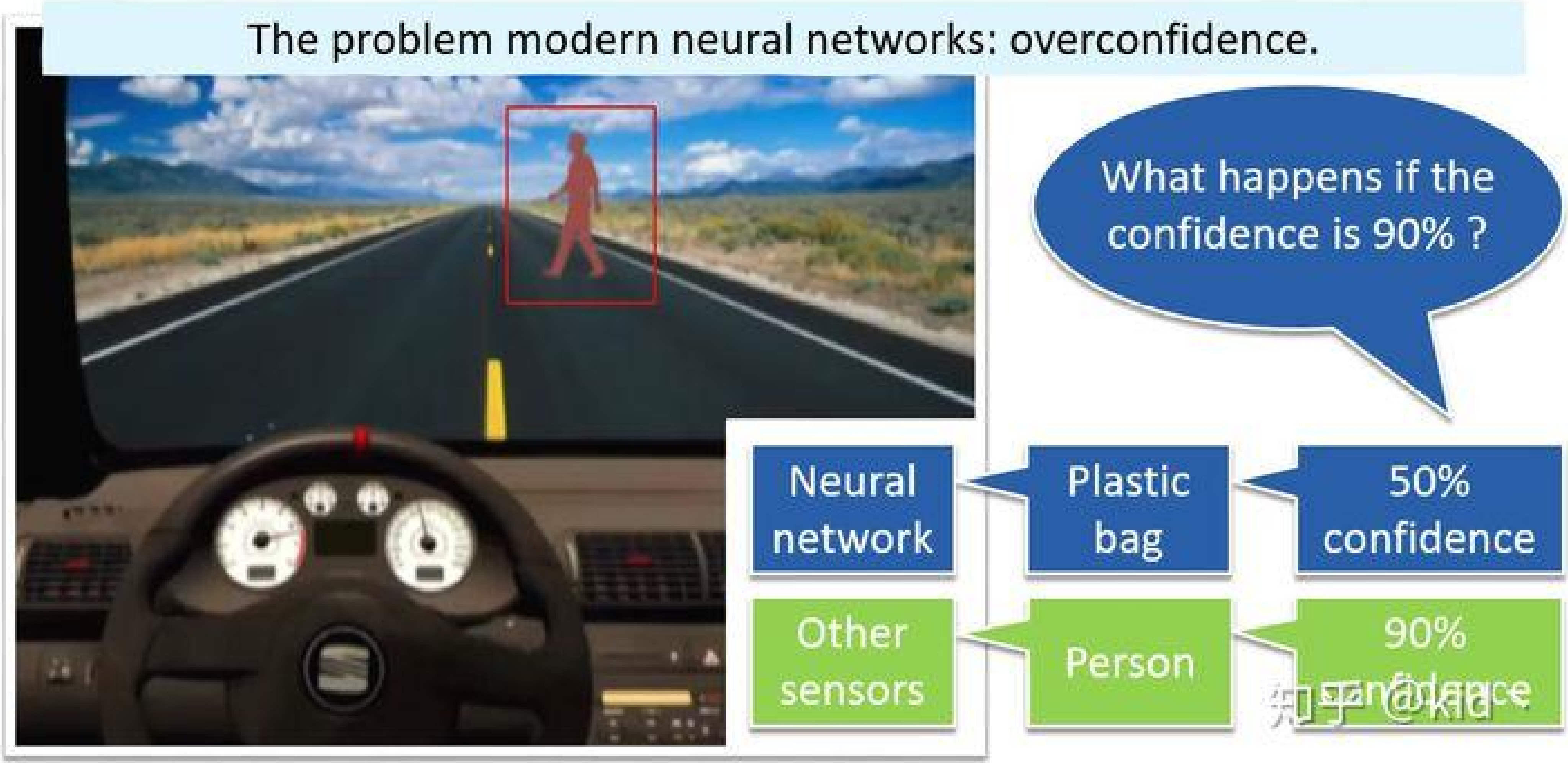
Chuan Guo, Geoff Pleiss, Yu Sun, Kilian Q. Weinberger
Cornell University

ICML 2017

The Problem of Modern Neural Networks: Overconfidence

The ResNet's accuracy is better but not match its confidence.

ResNet 101, Cifar 100
Samples with 80%-85% confidence


The problem modern neural networks: overconfidence.
What happens if the confidence is 90% ?
Neural network
Plastic bag
50% confidence
Other sensors
Person
90% confidence

Definitions

- perfect calibration:
- $$\mathrm{P}(Y = \mathrm{y}|\mathrm{Q} = \mathrm{q}) = \mathrm{q}$$

Cannot be computed using finitely many samples since Q is a continuous random variable

- miscalibration:
- $$E[|\mathrm{P}(Y = \mathrm{y}|\mathrm{Q} = \mathrm{q}) - q|]$$

approximations

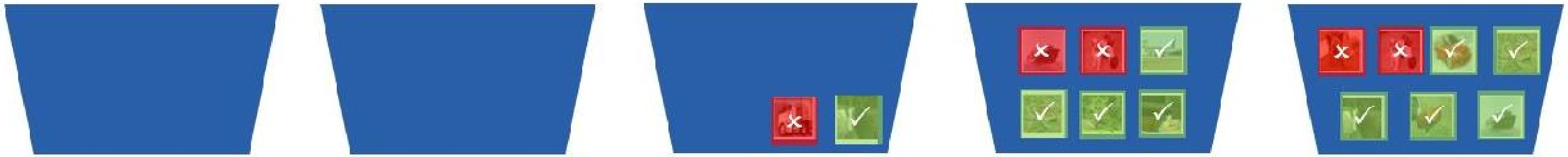
- ECE(expected calibration error):
- $$ECE = \sum_k \frac{b_k}{n} |acc(k) - conf(k)|$$

Definitions

- ECE(expected calibration error):
- $$ECE = \sum_k \frac{b_k}{n} |acc(k) - conf(k)|$$
- $$acc(k) = \frac{1}{|b_k|} \sum_{i \in b_k} 1(\widehat{y_i} = y_i)$$
- $$conf(k) = \frac{1}{|b_k|} \sum_{i \in b_k} (\widehat{q_i})$$

ECE(expected calibration error)

$0 \le \hat{p}(\mathbf{x}) < 0.2$	$0.2 \le \hat{p}(\mathbf{x}) < 0.4$	$0.4 \le \hat{p}(\mathbf{x}) < 0.6$	$0.6 \le \hat{p}(\mathbf{x}) < 0.8$	$0.8 \le \hat{p}(\mathbf{x}) < 1$

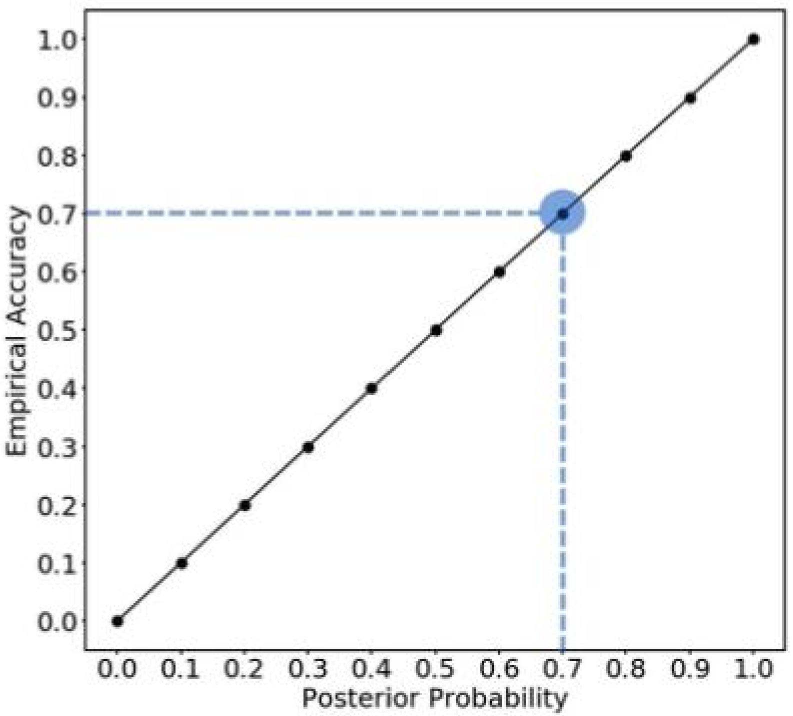


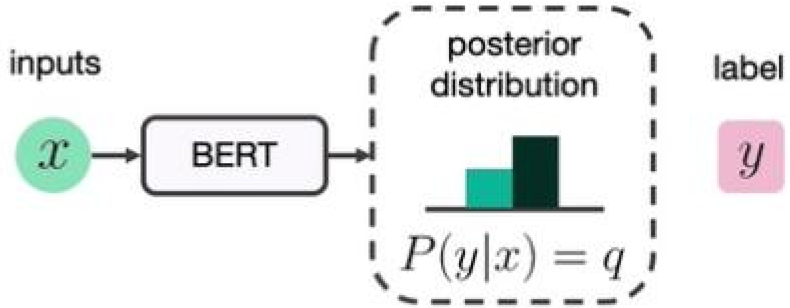
Avg conf:	0.55	0.74	0.86
— Accuracy:	0.50	0.67	0.71
Gap:	2 x 0.05	6 x 0.07	7 x 0.15

15

Expected Calibrated Error (ECE) = 0.11

Visualizing Calibration(perfectly calibrated)

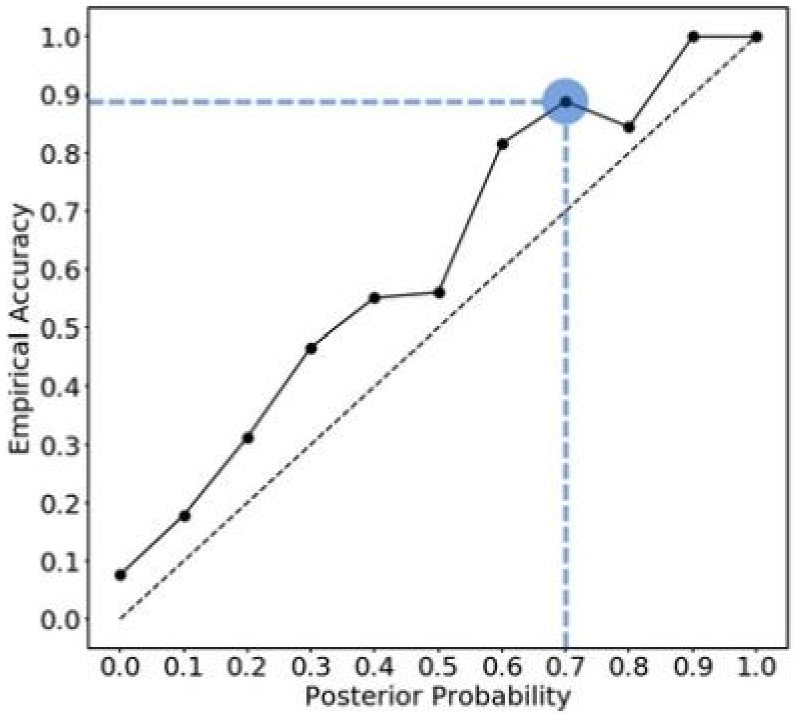


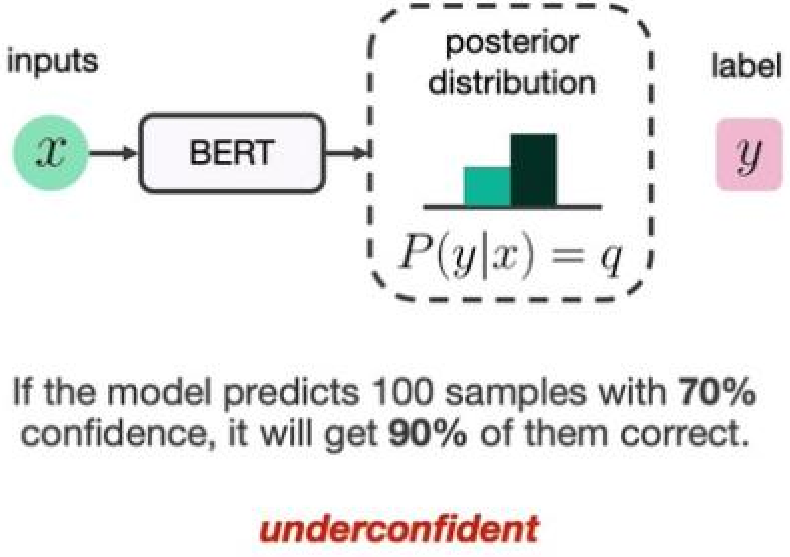


If the model predicts 100 samples with **70%** confidence, it will get **70%** of them correct.

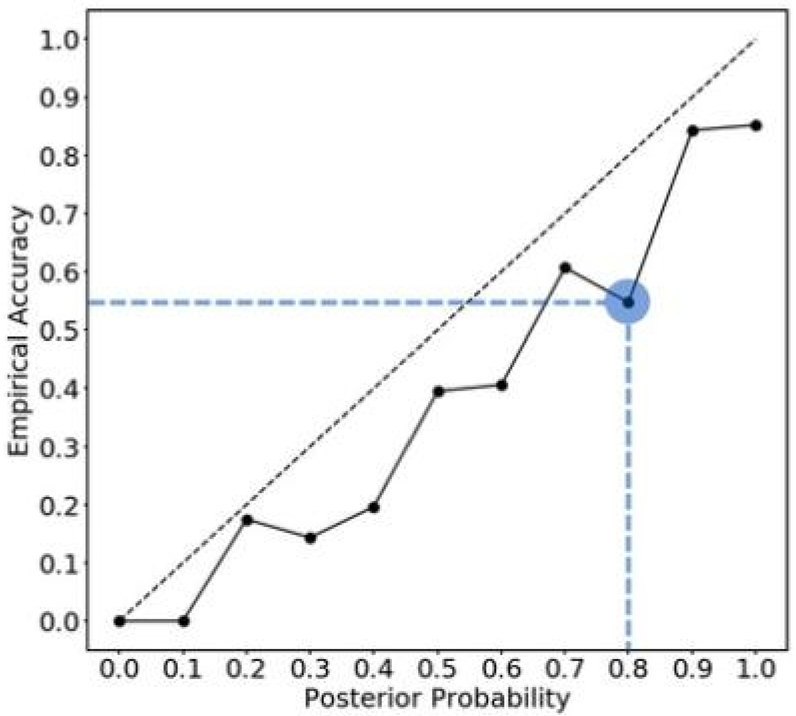
perfectly calibrated

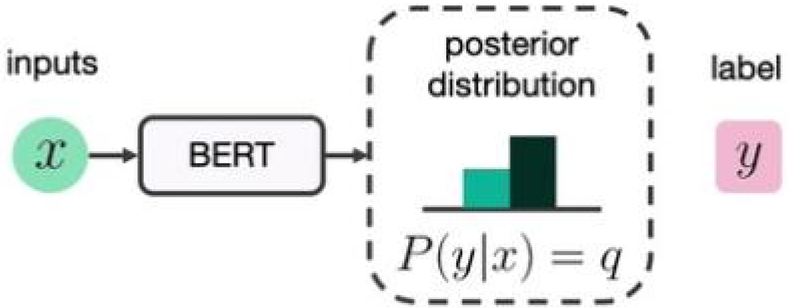
Visualizing Calibration(underconfident)





Visualizing Calibration(overconfident)

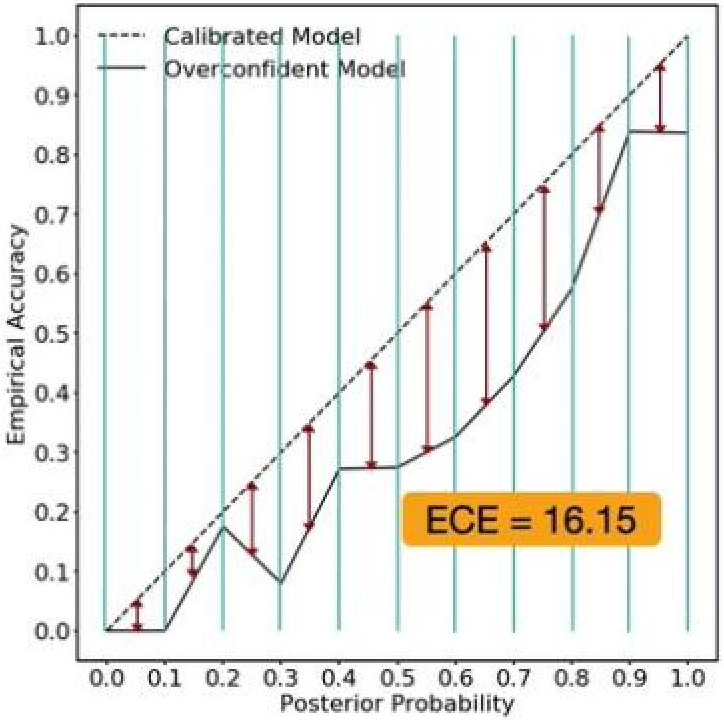




If the model predicts 100 samples with **80%** confidence, it will get **55%** of them correct.

overconfident

ECE



Computing **expected calibration error** (ECE):

1. Organize real-valued confidence scores into k disjoint, equally-sized bins (k=10)
2. Find the gap between the overconfident model and calibrated model
3. Sum over all residuals, weighted by the number of samples in each bin

We'll focus on **relative comparisons** of expected calibration error rather than ascribing significance to **absolute values**

Experiments

Tasks and Datasets

Tasks	Datasets	
	In-domain	Out-of-domain
Natural Language Inference	SNLI	MNLI
Paraphrase Detection	QQP	TwitterPPDB
Commonsense Reasoning	SWAG	HSWAG

Models

Model	Parameters	Architecture	Pre-trained
DA	382K	LSTM	✗
ESIM	4M	Bi-LSTM	✗
BERT	110M	Transformer	✓
RoBERTa	110M	Transformer	✓

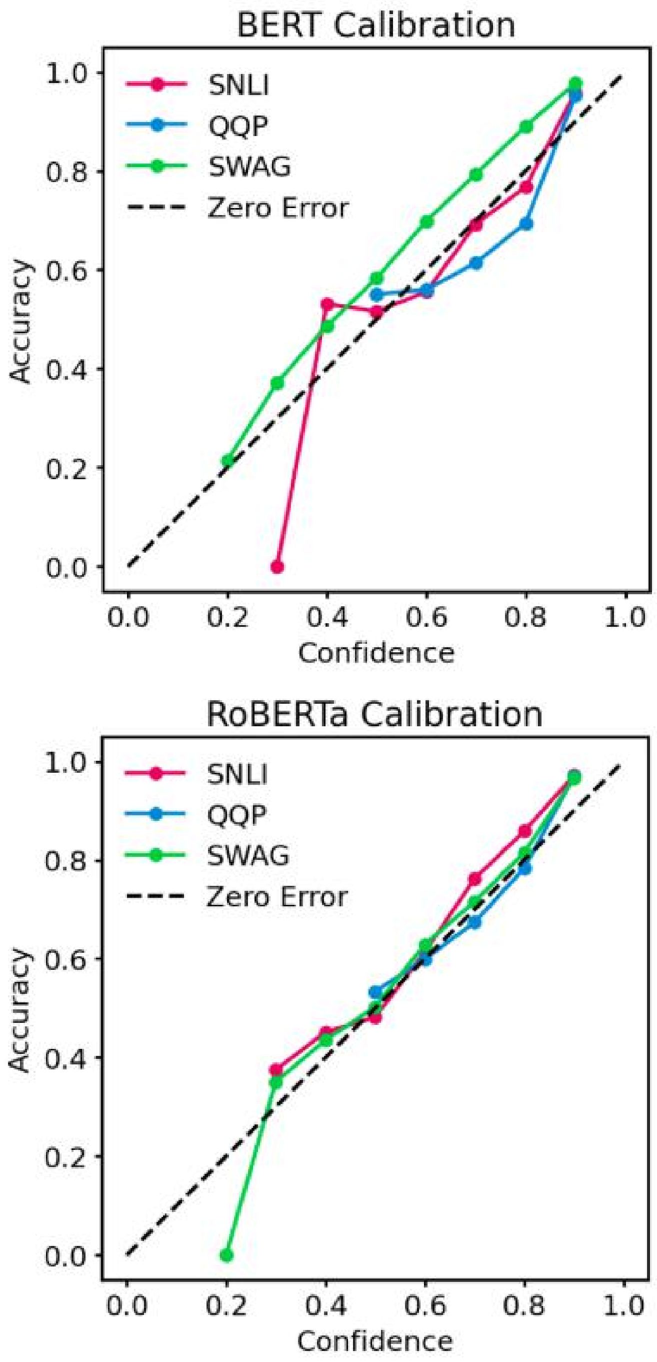
Experiments

- Out-of-the-box Calibration
  - the calibration error derived from evaluating a model on a dataset without using post-processing steps
- Post-hoc Calibration
  - temperature scaling
  - maximum likelihood estimation (MLE)
  - label smoothing (LS)

Out-of-the-box Calibration

Model	Accuracy		ECE	
	ID	OD	ID	OD
Task: SNLI/MNLI				
DA	84.63	57.12	**1.02**	8.79
ESIM	88.32	60.91	1.33	12.78
BERT	90.04	73.52	2.54	7.03
RoBERTa	**91.23**	**78.79**	1.93	**3.62**
Task: QQP/TwitterPPDB				
DA	85.85	83.36	3.37	9.79
ESIM	87.75	84.00	3.65	**8.38**
BERT	90.27	**87.63**	2.71	8.51
RoBERTa	**91.11**	86.72	**2.33**	9.55
Task: SWAG/HellaSWAG				
DA	46.80	32.48	5.98	40.37
ESIM	52.09	32.08	7.01	19.57
BERT	79.40	34.48	2.49	12.62
RoBERTa	**82.45**	**41.68**	**1.76**	**11.93**

- Non-pre-trained models exhibit an inverse relationship between complexity and calibration.
- pre-trained models are generally more accurate and calibrated.
- Using RoBERTa always improves in-domain calibration over BERT.



Post-hoc Calibration

- train the model with either MLE or LS using the in-domain training set
- use the in-domain development set to learn an optimal temperature T
- evaluate the model (scaled with T) on the in-domain and out-of-domain test sets

MLE & LS

- MLE
  - sharpen the posterior distribution around the gold label
- LS
  - minimize the KL divergence with the distribution
  - $1 - \alpha$ fraction of probability mass on the gold label
  - $\alpha/(|Y|-1)$ fraction of probability mass on each other label
  - where $\alpha \in (0, 1)$ is a hyperparameter.($\alpha = 0.1$)
  - For example, one-hot target [1, 0, 0] is transformed into [0.9, 0.05, 0.05] when $\alpha = 0.1$

temperature scaling

$$\frac{e^{\frac{z_y}{T}}}{\sum_{i=1}^{C} e^{\frac{z_i}{T}}}$$

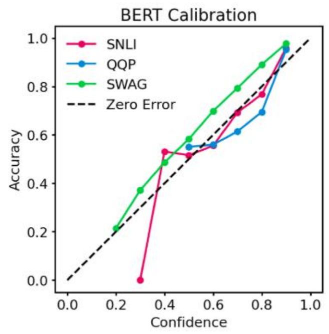
- z = [1 2 3 4]
  softmax(z) = [0.0321 0.0871 0.2369 0.6439]
- T=10
  z/T = [0.1 0.2 0.3 0.4]
  softmax(z/T) = [0.2138 0.2363 0.2612 0.2887]
- T=0.1
  z/T = [10 20 30 40]
  softmax(z/T) = [9.36e-14 2.06e-9 4.54e-5 1.00]

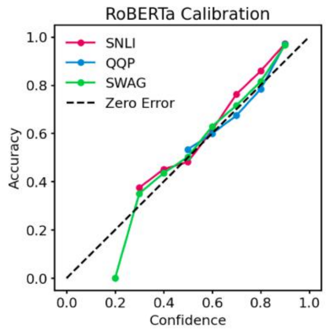
- As T → ∞, the probability approaches 1/K, which represents maximum uncertainty.
- With T = 1, we recover the original probability.
- As T → 0, the probability collapses to a point mass (i.e. 1)
- T does not change the maximum of the softmax function, the class prediction remains unchanged.
- temperature scaling does not affect the model’s accuracy.

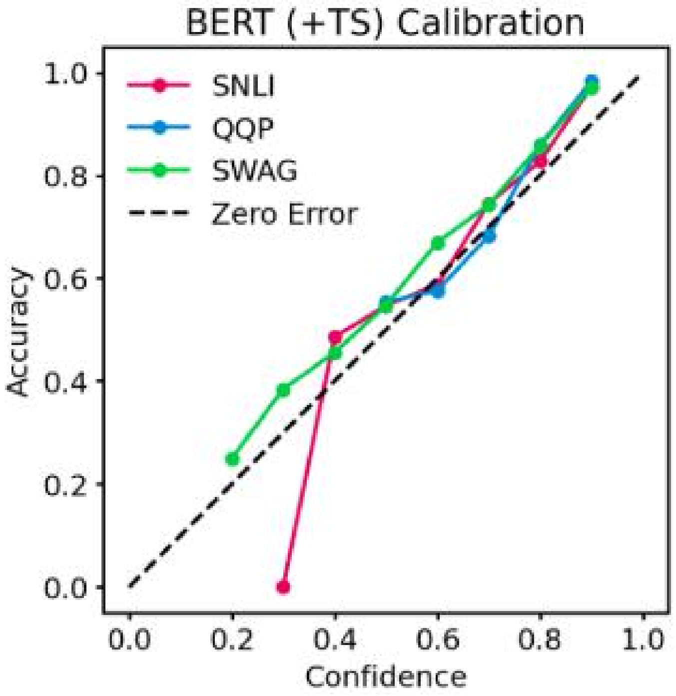
Post-hoc Calibration （ECE）

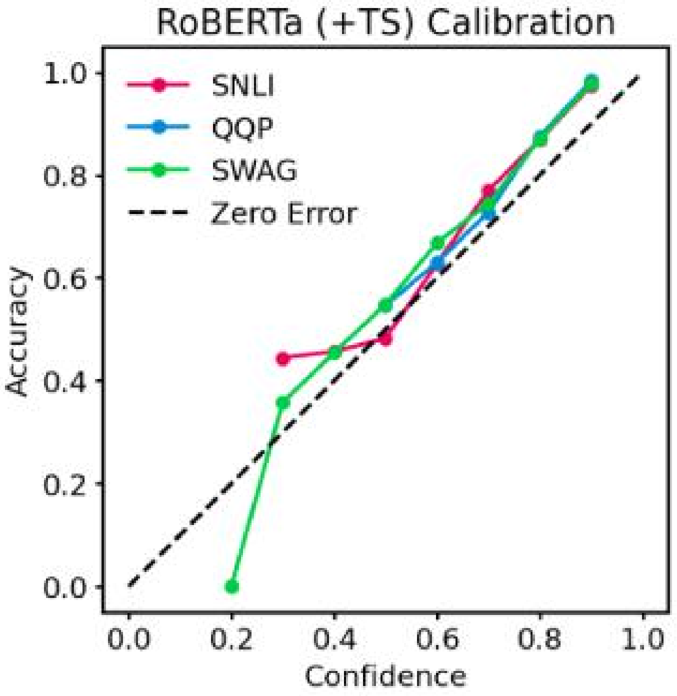
Method	In-Domain						Out-of-Domain					
	SNLI		QQP		SWAG		MNLI		TPPDB		HSWAG	
	MLE	LS	MLE	LS	MLE	LS	MLE	LS	MLE	LS	MLE	LS
Model: BERT												
Out-of-the-box	2.54	7.12	2.71	6.33	2.49	10.01	7.03	3.74	8.51	6.30	12.62	5.73
Temperature scaled	1.14	8.37	0.97	8.16	0.85	10.89	3.61	4.05	7.15	5.78	12.83	5.34
Model: RoBERTa												
Out-of-the-box	1.93	6.38	2.33	6.11	1.76	8.81	3.62	4.50	9.55	8.91	11.93	2.14
Temperature scaled	0.84	8.70	0.88	8.69	0.76	11.40	1.46	5.93	7.86	5.31	11.22	2.23

Post-hoc Calibration(In-domain)





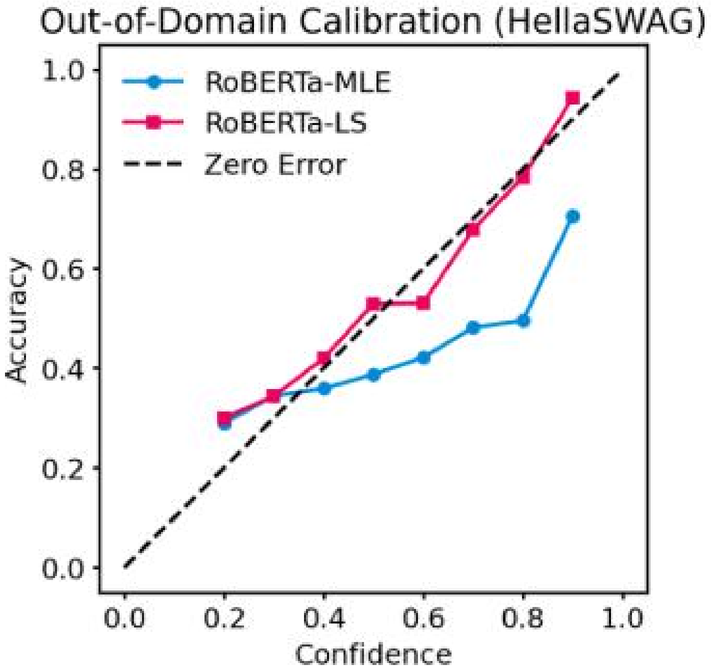




In-domain calibration of BERT and RoBERTa with temperature scaling (TS).

- Both models are much better calibrated than when used out-of-the-box
- BERT especially showing a large degree of improvement.

Post-hoc Calibration(Out-of-Domain)



Without seeing HellaSWAG samples during fine-tuning RoBERTa- LS achieves significantly lower calibration error than RoBERTa-MLE

Out-of-domain calibration of RoBERTa fine-tuned on SWAG with different learning objectives and used out-of-the-box on HellaSWAG

Post-hoc Calibration(Temperature Scaling)

Model	In-Domain			Out-of-Domain		
	SNLI	QQP	SWAG	MNLI	TPPDB	HSWAG
BERT	1.20	1.34	0.99	1.41	2.91	3.61
RoBERTa	1.16	1.39	1.10	1.25	2.79	2.77

Learned temperature scaling values for BERT and RoBERTa on in-domain and out-of-domain datasets.

- Values are obtained by line search with a granularity of 0.01.
- Evaluations are very fast as they only require rescaling cached logits.

Post-hoc Calibration

- MLE models with temperature scaling achieve low in-domain calibration error.
- out-of-domain, LS is generally more effective.

Conclusion

1. How are pre-trained Transformers calibrated relative to non-Transformer models when **trained and evaluated in-domain**?

Pre-trained transformers are generally more accurate *and* calibrated; RoBERTa > BERT

2. How are pre-trained Transformers calibrated relative to non-Transformer models when **trained in-domain** but **evaluated out-of-domain**?

Both BERT and RoBERTa show robustness out-of-domain, but still see high calibration errors.

3. How can we correct pre-trained Transformer calibration across both **in-domain** and **out-of-domain** settings?

Temp scaling is highly effective in-domain while label smoothing reduces error out-of-domain

谢谢！