Platt-Bin: Efficient Posterior Calibrated Training for NLP Classifiers

Rishabh Singh and **Shirin Goshtasbpour**
Department of Computer Science
ETH Zurich, Switzerland

Abstract

Modern NLP classifiers are known to return uncalibrated estimations of class posteriors. Existing methods for posterior calibration rescale the predicted probabilities but often have an adverse impact on final classification accuracy, thus leading to poorer generalization. We propose an end-to-end trained calibrator, Platt-Binning, that directly optimizes the objective while minimizing the difference between the predicted and empirical posterior probabilities. Our method leverages the sample efficiency of Platt scaling and the verification guarantees of histogram binning, thus not only reducing the calibration error but also improving task performance. In contrast to existing calibrators, we perform this efficient calibration during training. Empirical evaluation of benchmark NLP classification tasks echoes the efficacy of our proposal.

1 Introduction

Deep learning has proven to be tremendously attractive for researchers in fields such as physics, biology, and manufacturing, to name a few (Baldi et al., 2014; Anjos et al., 2015; Bergmann et al., 2014). However, these are fields in which representing model uncertainty is of crucial importance (Gal and Ghahramani, 2016). A common way to incorporate DNNs in other fields is to use the predictions of a trained classifier for decision making in a downstream task. In some cases the effectiveness of the decisions depends on a utility function and it is not enough to simply predict the most likely label for each example. What is needed instead is to quantify model uncertainty about the predictions. Despite promising performance in supervised learning benchmarks in terms of accuracy, DNNs are poor at quantifying predictive uncertainty, and tend to produce overconfident predictions. Overconfident incorrect predictions can be harmful or offensive in NLP applications (Amodei et al., 2016), hence proper uncertainty quantification is crucial in practice. Probabilistic uncertainty in machine learning translates to estimation of the probability mass function $p(y|\mathbf{x})$ by the model, where $\mathbf{x}$ is the input sample and y is a class label. Recent works have shown that state-of-the art structured prediction models are poorly calibrated. Therefore, blindly using the output of the softmax function output as the model uncertainty is misleading (Kumar and Sarawagi, 2019; Dong et al., 2018; Nguyen and O'Connor, 2015).

We are interested in calibrating the posterior estimates, i.e. we wish to get posterior probability estimations that reflect the true probability of the classes. The probability that a system outputs for an event should reflect the true frequency of that event: if an automated diagnosis system says 1,000 patients have cancer with probability 0.1, approximately 100 of them should indeed have cancer (Kumar et al., 2019). Even if the actual mechanism might be difficult to interpret, a calibrated model at least gives us a signal that it "knows what it doesn't know," thereby making these models easier to deploy in practice (Jiang et al., 2012). We define perfect calibration as follows.

$$\mathcal{P}(y|f(\mathbf{x})) = f(\mathbf{x})$$

where $f : \mathcal{X} \rightarrow \triangle_{K-1}$ is the probabilistic classifier that maps the samples $\mathbf{x} \in \mathcal{X}$ to the K-dimensional simplex. As majority of the current state-of-the art machine learning models, such as DNNs, do not output calibrated probabilities out of the box (Kuleshov et al., 2018), existing works rely on re-calibration methods that take the output of an uncalibrated model, and transform it into a calibrated probability. One way of addressing this is to use Scaling approaches for re-calibration such as Platt scaling (Platt et al., 1999), isotonic regression (Zadrozny and Elkan, 2002), and temperature scaling (Guo et al., 2017). These methods are widely used and require very few samples,

Findings of the Association for Computational Linguistics: ACL 2022, pages 3673 - 3684

however it is challenging to calibrate posterior estimates with sub-optimal binning schemes (Kumar et al., 2019). An alternative approach, histogram binning (Zadrozny and Elkan, 2001), outputs probabilities from a finite set. Histogram binning can produce a model that is calibrated, and unlike scaling methods we can measure its calibration error, but it is sample inefficient. In particular, the number of samples required for calibration scales linearly with the number of classes for which probability estimates need to be generated.

Irrespective of the choice of the calibration method, existing works generally calibrate the posterior distribution predicted from the classifier after training. These post-processing calibration methods re-learn an appropriate distribution from a held-out validation set and then apply it to an unseen test set. The fixed split of the data sets and insufficient number of samples for training the calibration function adversely affects the generalization of post-hoc calibrated classifiers and reduce their accuracy. In this paper we try to address some of the existing challenges in achieving apt calibration. In particular our contributions are:

- We propose a training technique that optimizes a classification objective for an NLP task by calibrating the posterior distribution while training.
- We leverage the advantages of both scaling and binning methods and propose a calibration method for NLP classification task which is both sample efficient and verifiable.
- We demonstrate how the proposed method not only calibrates but also improves the performance of benchmark NLP classification tasks.

2 Related Works

Model uncertainty estimation and posterior calibration is a topic of continued interest not only in the fields of machine learning and statistics, but also in meteorology (Bröcker, 2009), fairness (Liu et al., 2019), healthcare (Jiang et al., 2012), reinforcement learning (Malik et al., 2019), natural language processing (Card and Smith, 2018), speech recognition (Yu et al., 2011) and economics (Gneiting et al., 2007). In probabilistic models, the principal goal of estimation of the posterior $p(y|\mathbf{x})$ given a sample $\mathbf{x} \in \mathcal{X}$ and a label $y \in [K]$, is to assign low confidence to samples that were not explained well by the training data. One common way to calibrate multi-class posteriors after training the classifier $f : \mathcal{X} \to \mathbb{R}$ is to treat the problem as K one-vs-all binary problems. In this case, model uncertainty is quantified by normalizing the estimation of $p(y = k|f(\mathbf{x})_k)$ where $f(\mathbf{x})_k$ is the output score of the classifier for sample $\mathbf{x}$ and class k. Generalization of calibration tests with kernel methods can be found in (Widmann et al., 2019). Various binary calibration methods can be used to estimate the marginal posterior over a calibration dataset, ranging from parametric approaches (e.g. Platt scaling, temperature scaling, vector scaling (Platt et al., 1999; Guo et al., 2017)), to non-parametric methods (e.g. quantile or bayesian binning (Zadrozny and Elkan, 2001; Naeini et al., 2015), and isotonic regression (Zadrozny and Elkan, 2002).

Another way to reduce the problem to binary calibration is by estimating model accuracy conditioned on its confidence, $p(y = \hat{y}|\max_{k\in[K]} f(\mathbf{x})_k)$. Multi-class calibration aims to estimate the distribution of class labels conditioned on the estimated probability vector, $p(y|f(\mathbf{x}))$. In this case the sample complexity is exponential in the number of classes and therefore with large number of classes, the main challenge is to constrain the hypothesis space with regularization. Some of the proposed methods for this purpose are matrix scaling and Dirichlet scaling which both use linear models for estimation of $p(y = k|f(\mathbf{x}))$ (Guo et al., 2017; Kull et al., 2019), and MLP and order preserving functions (Rahimi et al., 2020a,b).

Another approach is to account for model uncertainty via bayesian models. In Bayesian Neural Networks (BNNs) the predictive uncertainty will naturally be high in regions where training data is scarce (MacKay, 1992). However, the marginalization of the weights in BNN is intractable in general. Consequently, following papers propose various approximations such as variational inference (VI) (Graves, 2011; Blundell et al., 2015). Although BNNs are theoretically proven to control the overconfidence of the model in unseen regions of data space (Kristiadi et al., 2020), they require expensive approximations which limit their application in most modern NLP architectures. For instance, in (Joo et al., 2020) the authors model the distribution on posterior probability using a Dirichlet prior distribution and variational inference. MCDropout is

a variational approximation of Gaussian processes that avoids explicit modeling of the posterior distribution (Gal and Ghahramani, 2016). Both of these methods require modification of training of the network.

In NLP, tasks with structured outputs posterior calibration are particularly challenging. This is because the number of classes are exponentially large and estimation of every posterior density or marginal posterior density is not possible. Previous works such as (Jung et al., 2020; Nguyen and O'Connor, 2015) propose to use the downstream task with small number of classes to perform calibration and estimation of the calibration error. In structured prediction models, calibration is also important for the generation of the structured outputs as the decoding algorithm relies on the posterior estimates to efficiently search through the space of sequences. However, estimation of the sequence calibration error and its correction is intractable. To cope with this problem, approximate calibration methods using a set of interesting events and feature based calibration are proposed in (Kuleshov and Liang, 2015; Jagannatha and Yu, 2020) and an alternative calibration error estimator was proposed using sequence precision scoring function BLEU in (Kumar and Sarawagi, 2019). We are considering the first class of problems and leave the structured calibration to future work.

3 Method

In general, NLP classifiers work by first predicting a posterior probability distribution over all classes and then selecting the class with the largest estimated probability. However, these models are often poorly calibrated. Existing calibration methods re-learn an appropriate distribution from a held-out validation set and then apply it to an unseen test set which degrades the model performance. Alternatively, we can dynamically estimate the required statistics for calibration from the train set during training iterations, thereby minimizing cross-entropy as well as the calibration error as a multi-task setup (Jung et al., 2020). Given a training set $D = \{(x_1, y_1)..(x_n, y_n)\}$, where x_i is an n-dimensional vector of input features and y_i is a K-dimensional one-hot vector corresponding to its true label (with K classes), we minimize the loss L_{train}:

$$L_{train} = L_{class} + \lambda L_{cal} \quad (1)$$

Here L_{class} is the classification loss (for eg. cross-entropy) based on the predicted probability p_{ik} updated during training for sample i and class k:

$$L_{class} = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} log(p_{ik})$$

L_{cal} is the calibration loss which acts as a regularizer. It essentially tries to minimize the difference between the updated probability p and true posterior probabilities q via a distance function d (eg. mean squared error, KL-divergence, etc.):

$$L_{cal} = \sum_{i=1}^{N}\sum_{k=1}^{K} d(p_{ik}, q_{ik})$$

One crucial step here is to estimate the empirical probability q, which can be done by histogram binning method. Here, we measure the ratio of true labels for each bin split by the predicted posterior p from each update. This refers to *CalEmpProb()* function in algorithm 1. We store the results in Empirical Probability Matrix $Q \in \mathbb{R}^{B\times K}$, where B is the number of bins used for each posterior dimension. Histogram binning outputs probabilities from a finite set. Unlike scaling methods, it can produce a model that is calibrated and measure its calibration error. However, the number of samples required to calibrate scales linearly with the number of distinct probabilities B the model can output which can be large in the multi-class setting (Naeini et al., 2014).

In this work, we propose an adaptive binning method that circumvents this bottleneck. We leverage the sample efficiency of Platt scaling (Platt et al., 1999) and the verification guarantees of histogram binning (Zadrozny and Elkan, 2001) by defining the *Platt-Binning Calibrator*. The problem with scaling methods is we cannot estimate their calibration error. The upside of scaling methods is that if the function family has at least one function that can achieve calibration error ϵ they require $O(1/\epsilon^2)$ samples to reach calibration error ϵ, while histogram binning requires $O(B/\epsilon^2)$ samples. *Platt-Binning Calibrator* facilitates estimation of calibration error while being sample-efficient at the same time.

Platt scaling calibrator: Since most modern deep learning classifiers do not output calibrated probabilities out of the box, recalibration methods take

the output of an uncalibrated model, and transform it into a calibrated probability. That is, given a trained model $f : \mathcal{X} \rightarrow [0, 1]$, let $\mathbf{z} = f(\mathbf{x})$. We are given recalibration data $T = \{(\mathbf{z}_i, y_i)\}_{i=1}^{n}$ corresponding to model logits and the labels, and we wish to learn a calibrator $g : [0, 1] \rightarrow [0, 1]$ such that $g \circ f$ is well-calibrated. Conventional Scaling methods, for example Platt scaling, output a function g:

$$g = \arg\min_{g \in G} \sum_{(\mathbf{z},y)\in T} l(g(z), y)$$

where G is a the hypothesis class, $g \in G$ is differentiable, and l is a loss function, for example the log-loss or mean-squared error. The advantage of such methods is that they converge very quickly since they only fit a small number of parameters. ***Histogram binning calibrator***, on the other hand, constructs a set of bins that partitions $[0, 1]$ via a binning scheme. A binning scheme $\hat{B}$ of size B is a set of B intervals $I_1, ...I_B$ that partitions $[0, 1]$. We use the notation σ to denote the softmax function. Given $p = \sigma(\mathbf{z})_k \in [0, 1]$, let $\beta(z) = j$, where j is the interval that p lands in ($p \in I_j$). The binning scheme, $\hat{B}$ typically corresponds to choosing bins of equal widths (called equal width binning) or so that each bin contains an equal number of $\mathbf{z}_i$ values in the calibration dataset (called uniform mass binning). Histogram binning then outputs the average y_i value in each bin.

Platt-Binning Calibrator builds at the intersection of the above two methods. Given a recalibration data T of size n, *Platt-Binning Calibrator* outputs $\hat{g}_\beta$ such that $\hat{g}_\beta \circ f$ has a low calibration error by using the following procedure:
Step 1: Select g:

$$g = \arg\min_{g \in G} \sum_{(z,y)\in T} (y - g(z))^2 \qquad (2)$$

Step 2: Choose the bins so that an equal number of $g(z_i)$ in T land in each bin b_j for each $j \in 1, ..., B$

$$\text{ECE} = \frac{1}{K} \sum_{k=1}^{K} \sum_{b=1}^{B} \frac{N_{kb}}{N_k} |Q_{bk} - \bar{p}_{bk}|$$

where $\bar{p}_{bk}$ is the average posterior estimate for class k for samples in b-th bin. N_{kb} and N_k are the number of samples of class k assigned to bin b and in total, respectively. Contrary to equal-width binning, uniform-mass binning is a well-balanced binning scheme with guarantees on error bounds of estimated Expected Calibration Error, ECE (Kumar et al., 2019).
Step 3: Discretize g, by outputting the average g value in each bin. Let $\mu(S) = \frac{1}{|S|} \sum_{s \in S} s$ denote the mean of a set of values S. We set $\hat{g}_\beta(z) = \mu(\beta(g(z)))$- we output the mean value of the bins that $g(z)$ falls in.

The motivation behind our method is that the g values in each bin are in a narrower range than the label values y, so when we take the average we incur lower estimation error. If G is well chosen, our method requires $O(\frac{1}{\epsilon^2} + B)$samples to achieve calibration error ϵ instead of $O(\frac{B}{\epsilon^2})$ samples for histogram binning. All these steps are performed during training as explained in the pseudo-code in Algorithm 1. To the best of our knowledge, such a formulation is novel among existing calibrators that tackle the problem during training. Also, the whole approach is the first to be utilised to calibrate classifiers in the NLP domain. In the following section we prove the efficacy of our method by carrying out extensive evaluation of the performance of pre-trained transformer models such as BERT (Devlin et al., 2019) on simple multi-class text classification tasks. Our motivation comes from the analysis in (Desai and Durrett, 2020) which shows that pre-trained models are significantly better calibrated when used out-of-the-box.

4 Experiments

In the experiments we fine-tune the parameters on pre-trained BERT classifier using the regularized loss in equation (1). We compare our method to the following baselines:

- **MLE** is the baseline with maximum likelihood training without calibration where we simply report the results of vanilla BERT classifier on the chosen tasks.
- **Platt scaling** (posPS) is a post-hoc calibration method where we calibrate the posterior estimations of MLE classifier using Platt scaling (Platt et al., 1999). Formally, the parameters of the calibration functions $g(\mathbf{z}; \mathbf{W}, \mathbf{b}) = NN(\mathbf{W} \cdot \sigma(\mathbf{z}) + \mathbf{b})$ is fit to the validation dataset. Here, *NN* refers to a neural network with the component-wise logistic function. Model is fit using one-vs-all binarization of the classification task. Instead of the estimated posterior-

Algorithm 1 Platt-Binning Calibrated Training

Input: Train set D, j^{th} bin b_j, Set of all bins b, Number of Classes K, Number of epochs e, Learning rate η, Update period u
Output: Model Parameters Θ
Let Q : Empirical Probability Matrix $\in \mathbb{R}^{B\times K}$
Random initialization of Θ
for $i \in$ *{1,2,3, ...e}* **do**
  Break D into random mini-batches m
  for m *from* D **do**
    if i *mod* $u == 0$ **then**
      $\hat{p}(x) = \max_k \sigma(\Theta, D)_k$,
      $\forall x \in D$.
      $\hat{y} = \arg\max_k \sigma(\Theta, D)_k$,
      $\forall x \in D$.
      Select g using equation 2.
      Uniform-mass binning over $g(p_i)$.
      Discretize g: $\hat{g}_\beta(p_i) = \mu[\beta(g(p_i))]$

      $Q \leftarrow$ CalEmpProb($\hat{p}$;b_j)
    end
    $\Theta \leftarrow \Theta - \eta\nabla_\Theta L_{train}(\Theta, \hat{g}_\beta(p_i), b)$
  end
end

$\sigma(f(\mathbf{x}))_k$ for class k- we return the calibrated value $g(f(\mathbf{x}))$ as the class probability. Despite its simplicity this method is competitive with the more complex methods when implemented post-hoc (Guo et al., 2017).

- **PosCal** end-to-end training calibration using histogram binning (Jung et al., 2020). In this method we have a nested training procedure where in the outer loop we fit a histogram binning scheme with fix widths to each dimension of the posterior estimates of the BERT model. We use Q_{bk}- the ratio of samples of kth class that were assigned to bth bin- as the empirical probability distribution q. In the inner loop we perform the ordinary training iterations over mini-batches of training dataset with cross-entropy loss and regularization term in equation (1) using KL-divergence between softmax output and the estimated empirical distribution.

$$L_{cal} = \sum_{i=1}^{N}\sum_{k=1}^{K} \log \frac{\sigma(\mathbf{z}_i)_k}{Q_{\text{bin}(z_{ik})k}}$$

where bin(.) returns the index of bin assigned to its input. In the experiments we used $\lambda = 1.0$, 10 bin for discretisation of q and we update Q after every training epoch.

We test the baselines and our method on the benchmark on NLP classification tasks: xSLUE (Kang and Hovy, 2019). xSLUE contains classification benchmark on different types of styles such as a level of humor, formality and even demographics of authors. We train our method with two types of calibrators: in the first calibration task we train a calibrator for the most confident prediction of the classifier and call this version *plattbintop* (**PBtop**). The pseudocode of this version is illustrated in algorithm (1). In the second version we train a separate Platt scaler and histogram binning for each class in a one-vs-all manner and we call this version of calibration *plattbin* (**PB**). While this version is exactly the same as *plattbintop* for binary tasks, it results in a very different solution for tasks with $K > 2$. The pseudocode of this version is omitted due to being mainly similar to the other version with one additional loop over the classes at line 7 of algorithm (1) and conversion of label y and $\hat{y}$ to one-vs-all binary labels. We report task accuracy, F1 score and ECE as the evaluation metrics.

5 Results and Discussion

Table 1 shows task performance and calibration error on xSLUE benchmark datasets. In general, our method outperforms MLE, Poscal and posPS on more than 50% of the datasets, in terms of both model performance and calibration error. For the rest of the datasets, our method gives competitive results. In seven out of nine cases, we reduce the calibration error ECE as compared to PosCal. In cases such as *DailyDialog*, *SentiTreeBank* and *ShortHumor*, the achieved reduction in ECE as compared to all baselines is significant. Note that this reduction has not compromised the model performance. In fact, cases like *SentiTreeBank* and *ShortRomance* even witness a significant improvement in the performance of the model when ECE is reduced. These observations prove the efficacy of our method in maintaining a perfect balance between model performance and model uncertainity- a testimony of an ideal calibrator. Post-hoc methods such as posPS might achieve lower calibration error on a couple of datasets, but they fail to attain competitive performance in terms of accuracy. Similarly, in-training methods like PosCal tend to achieve higher accuracy but fail to be consistent in

Dataset	Accuracy					F1 score					ECE				
	MLE	PosCal	posPS	PB	PBtop	MLE	PosCal	posPS	PB	PBtop	MLE	PosCal	posPS	PB	PBtop
DailyDialog	84.8	84.1	84.8	**84.9**	83.7	29.4	29.9	28.4	29.8	**30.6**	16.5	13.2	10.5	**9.6**	11.5
HateOffensive	91.5	94.4	93.4	92.9	**95.9**	84.1	86.5	86.8	85.0	**91**	13.6	8.3	3.9	12.6	**3.8**
SarcasmGhosh	54.4	54.4	54.4	**54.5**	54.5	42.5	42.5	42.5	**43.0**	42.6	91.1	91.1	89.7	**89.5**	90.9
SentiTreeBank	94.6	93.9	94.5	95.4	**95.8**	94.6	93.9	94.5	95.4	**95.8**	9.6	8.0	7.1	**4.8**	5.1
ShortHumor	95.4	95.0	95.5	95.7	**95.8**	94.4	95.0	95.5	95.7	**95.8**	7.9	7.3	4.6	5.9	**3.6**
ShortRomance	99.9	96.0	99	**99.9**	98	98.9	95.9	98.9	**99.1**	97.9	3.0	7.1	3.0	**2.3**	2.5
StanfordPoliteness	67.9	56.1	67.9	**68.1**	66.8	68.0	53.5	66.9	**68.2**	65.6	22.3	59.1	**8.1**	23.0	24.4
TroFi	77.5	**78.8**	77.5	75.3	74	75.9	**77.7**	76.2	74.7	73.5	18.4	24.4	**16.7**	21.8	23.6
VUA	80.6	81.6	81.2	80.8	**81.7**	77.4	**78.5**	77.5	73.7	74.6	28.5	14.7	16.5	12.1	**9.9**

Table 1: Comparison of Model performance and Calibration error on different benchmark datasets. MLE: Maximum Likelihood; PosCal: Posterior Calibrated Training with Histogram Binning; posPS: post-hoc calibration with Platt scaling; PB: Platt-Binning Method; PBtop: PB over max(softmax(logits)). Our method (PB or PBtop) achieves better balance among the three metrics reported.

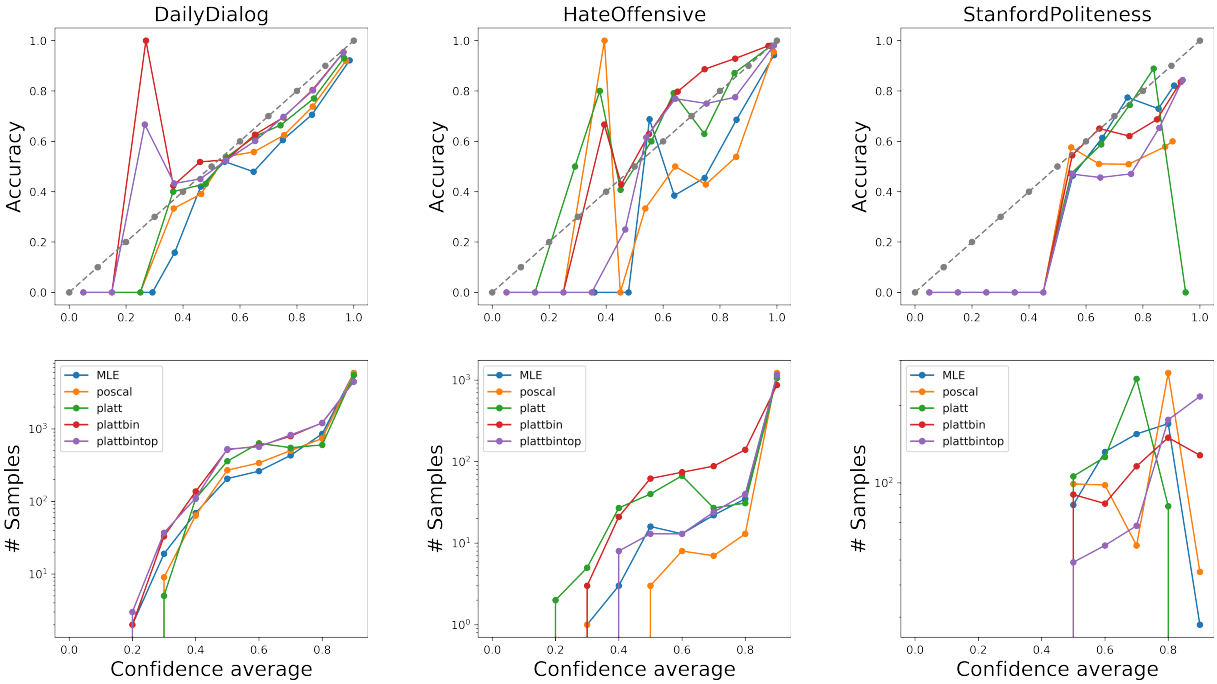


Figure 1: Calibration plots: (top) accuracy vs average confidence, (bottom) number of samples per bin vs average confidence

reducing calibration error. Our proposed method (PB or PBtop) hits the sweet-spot between the two extremes and is shown to achieve better results than baselines: highest accuracy except for TroFi, highest F1 score except for TroFi and VUA and lowest ECE except for TroFi and stanfordpoliteness (Table 1).

We now analyse how our method behaves in comparison to MLE at sample level during test time. Table 2 shows a detailed analysis of misclassification made by MLE and Platt-Binning (PB). We see that both the methods have almost comparable performance in columns $A1$ and $A2$, with $A2$ being slightly higher. As such, the number of samples for which MLE and PB gave different predictions (column M) is actually a small fraction of the total number of test samples used of evaluation of the methods (column $Test$). We further analyse the number of samples where MLE gave correct predictions while PB failed to do so (column $P1$) and vice-versa (column $P2$). In 8 out of 9 datasets, PB demonstrates superior or similar performance (P2 $\geq$ P1). The difference is insignificant compared

Data	Test	M	P1	P2	A1	A2
DailyDialog	7740	475	244	**292**	84.7	**84.9**
HateOffensive	1255	93	32	**50**	91.4	**92.9**
SarcasmGhosh	2000	0	0	0	54.4	54.4
SentiTreeBank	1749	73	29	**44**	94.5	**95.4**
ShortHumor	2256	93	44	**49**	95.4	**95.6**
ShortRomance	100	0	0	0	99.9	99.9
StanfordPoliteness	567	75	37	**38**	67.9	**68.1**
TroFi	227	41	23	18	77.5	75.3
VUA	5873	958	472	**486**	80.6	**80.9**

Table 2: Comparison of model performance at test time between MLE and PB. **Test**: Number of test samples, **M**: No. of test samples for which MLE and PLatt-Binning (PB) gave different predictions, **P1**: No. of samples correctly classified by MLE but misclassified by PB , **P2**: No. of samples correctly classified by PB but misclassified by MLE, **A1**: Accuracy of MLE, **A2**: Accuracy of PB

to the total size of the test set for the reverse scenario. This quantitative analysis reinstates that our method, PB, has better model performance at test time, thereby establishing that it generalizes well while reducing calibration error.

We extend the discussion above by analysing qualitative results in Table 3. We consider three datasets- a two-class classification task *StanfordPoliteness*, a three-class classification task *HateOffensive* and a multi-class classification task ($K > 3$) *DailyDialog*, and include few test samples where MLE and PB disagreed on the predictions. The corresponding $\hat{p}$ along with the true label is also depicted.

In the first two cases from *StanfordPoliteness* dataset, the level of politeness (e.g., "Hey!" in S1) or arrogance (e.g., "What?" in S2) indicated on phrases is not captured well by MLE, so it predicts the incorrect label while PB gives a correct prediction. However, for the rest two cases, MLE gives confident correct predictions taking into account phrases such as "like" in S1 or a slightly difficult example in S2 but PB fails (only slightly in S2 though) to give correct predictions. Arguing on similar lines for the multi-class case, we witness cases where MLE fails to classify correctly (eg. S1 and S2 in *HateOffensive*) but PB gives highly confident predictions and vice-versa. From our manual investigation above, we find that statistical knowledge about posterior probability helps correct $\hat{p}$ while training PB, so making $\hat{p}$ switch its prediction. For further analysis, we provide more examples in Appendix **??**.

In Figure 1 we show the calibration plots for three datasets: *DailyDialog*, *HateOffensive*, and *StanfordPoliteness*. We divide test samples according to the most confident estimated posterior into 10 bins. We plot the accuracy of the classifier versus the average classification confidence in each one of the bins in the top row. We also plot the number of samples in each calibration bin versus the classification confidence in the bottom row. Ideally, a calibrated classifier would assign a probability to the top class that is equivalent to its accuracy. Therefore, the accuracy-confidence curve of a calibrated classifier is close to the dashed grey curve in the top row. When Platt-bin and Platt-bin-top are further away from the calibration line it is because the number of samples in corresponding bins are low or even 0 in some cases. The bins with 0 samples in them can be ignored as they don't play a role in the classifier predictions.

However, the distance of the curves is not enough to determine model calibration as most of the samples are assigned to the bin with highest estimated posterior. Thus, correcting the calibration error in the bins with more samples is more effective in improving the expected calibration error. *Platt-Binning* and *Platt-Binning-Top* algorithms increase the number of samples with lower classification confidence in all three of the illustrated tasks, while in comparison to *MLE* with no regularization they only reduce classification accuracy by a negligible amount and even increase the accuracy for *HateOffensive* task. Although, the classifier become visibily underconfident in *HateOffensive* task where post-hoc Platt scaling has a more calibrated output. While the ECE doesn't improve in *StanfordPoliteness*, *Platt-Binning* algorithm doesn't increase the ECE as much as *PosCal* regularization. We conjecture that such a behavior is demonstrated due to better sample efficiency of our algorithm.

We conclude our analysis by observing the effect of two important parameters to this discussion- B: number of histogram bins used for calibration, and λ: strength of the regularization. Figure 2 shows how calibration error (ECE) vary when the number of bins B is varied as $\{10, ...100\}$. We see that the calibration error of all the methods have an increasing trend as B is increased. One plausible

Data	Sentence	True Label	$\hat{p}$ (MLE)	$\hat{p}$ (PB)	MLE → PB
DailyDialog	**S1:** Really ? What did you get one for ?	surprise	0.17	0.60	INCOR→COR
	S2: To hell with you . The accident was your fault	anger	0.14	0.41	INCOR→COR
	S1: I might just ! Enjoy your stupid game !	anger	0.41	0.36	COR→INCOR
	S2: Yeah . We rolled out the red carpet to welcome him home .	noemotion	0.96	0.37	COR→INCOR
HateOffensive	**S1:** @HBergHattie @snkscoyote I wonder if the progs didn't relegate young black men to the ghettos to keep them away from harry reid's friends.	neither	0.02	0.91	INCOR→COR
	S2: Every spic cop in #LosAngeles is loyal to the #LatinKin	hate	0.002	0.65	INCOR→COR
	S1:"Our people". Now is the time for the Aryan race 2 stand up and say "no more". Before the mongerls turn the world into a ghetto slum.	hate	0.95	0.37	COR→INCOR
	S2: #RebelScience .....is using an ACTUAL WOMAN as a genetic engineering lab for "all natural clones"..... or something..... #faggot #ro	hate	0.98	0.04	COR→INCOR
StanfordPoliteness	**S1:** Hey, long time no seeing! How's stuff?	polite	0.16	0.63	INCOR→COR
	S2: What user list? The one I linked to?	impolite	0.34	0.52	INCOR→COR
	S1:I like the first shot. Are those doghouses?	polite	0.68	0.24	COR→INCOR
	S2: I usually just boil water and then drink but I think it won't help here. Does it?	impolite	0.68	0.48	COR→INCOR

Table 3: Predicted $\hat{p}$ of true label from MLE and PB with corresponding sentences in D-Dialog, H-Offensive and S-Polite dataset. Provided examples contrast the predictions between MLE and PB for qualitative analysis.

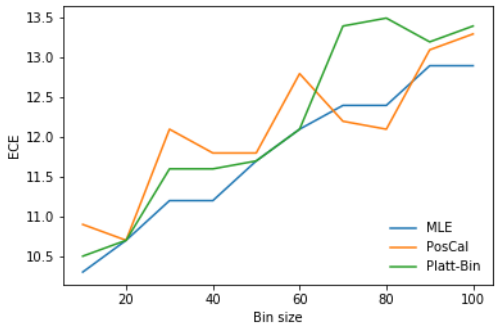


Figure 2: Effect of number of histogram bins used for calibration on the calibration error

explanation can be that as we increase the number of bins, we don't have enough samples per bin to estimate the empirical probabilities accurately. Since calibrated probabilities are used as an estimation of the true probabilities of the classes in case of PosCal and PB, it adds to the error if they are estimated wrongly. Thus, smaller number of bins is preferred, and as evident in Fig. 2, *PB* achieves lower ECE than *PosCal* when number of bins is low. The accuracy and F1 scores do not vary much with the number of the bins. Similarly, the performance is not impacted significantly by variations in the value of λ (see Appendix **??**)

6 Conclusion

In this work we proposed a simple yet effective method called Platt-Binning calibrator for better posterior calibration. Our method has theoretically lower sample complexity than histogram binning, giving us the best of scaling and binning methods. And unlike the existing post-processing calibration methods, Platt-Binning directly penalizes the difference between the predicted and the true (empirical) posterior probabilities dynamically over the training steps. Our empirical analysis corroborates that Platt-Binning can not only reduce the calibration error but also increase the task performance on the classification benchmarks. For tasks where the reduction in calibration error is low, our method maintains the performance of the model instead of degrading it as seen for other existing calibrators. Moreover, our method can be extended to any classification model as an additional component in the loss function, thus jointly optimised during training. There are many exciting avenues for future works in this regard. It will be interesting to assess how our method can provide advantages in the scenarios of domain adaptation and transfer learning. Moreover, exploring alternatives to the model family G from which estimate $\hat{g}$ is considered can be a direction of improvement. Lastly, optimizing the overall method for huge datasets can be

an essential extension. Our method may also assist in analysing the bias and fairness aspects of the predictions made by NLP classifiers. This can facilitate ethical deployment of NLP models for real-world applications.

Acknowledgements

Shirin Goshtasbpour is supported by funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 813999 for this project.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Ofélia Anjos, Carla Iglesias, Fátima Peres, Javier Martínez, Ángela García, and Javier Taboada. 2015. Neural networks applied to discriminate botanical origin of honeys. *Food chemistry*, 175:128–136.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. 2014. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1):1–9.

Sören Bergmann, Sören Stelzer, and Steffen Strassburger. 2014. On the use of artificial neural networks in simulation-based manufacturing control. *Journal of Simulation*, 8(1):76–90.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.

Jochen Bröcker. 2009. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 135(643):1512–1519.

Dallas Card and Noah A Smith. 2018. The importance of calibration for estimating proportions from annotations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1636–1646.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Li Dong, Chris Quirk, and Mirella Lapata. 2018. Confidence modeling for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 743–753.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. 2007. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268.

Alex Graves. 2011. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Abhyuday Jagannatha and Hong Yu. 2020. Calibrating structured output predictors for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2078–2092.

Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. 2012. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274.

Taejong Joo, Uijung Chung, and Min-Gwan Seo. 2020. Being bayesian about categorical probability. In *International Conference on Machine Learning*, pages 4950–4961. PMLR.

Taehee Jung, Dongyeop Kang, Hua Cheng, Lucas Mentch, and Thomas Schaaf. 2020. Posterior calibrated training on sentence classification tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2723–2730.

Dongyeop Kang and Eduard Hovy. 2019. xslue: A benchmark and analysis platform for cross-style language understanding and evaluation.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. 2020. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International conference on machine learning*, pages 5436–5446. PMLR.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. 2018. Accurate uncertainties for deep learning using calibrated regression. In *International conference on machine learning*, pages 2796–2804. PMLR.

Volodymyr Kuleshov and Percy S Liang. 2015. Calibrated structured prediction. In *Advances in Neural Information Processing Systems*, pages 3474–3482.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. 2019. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, pages 12316–12326.

Ananya Kumar, Percy S Liang, and Tengyu Ma. 2019. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems*, pages 3792–3803.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.

Lydia T Liu, Max Simchowitz, and Moritz Hardt. 2019. The implicit fairness criterion of unconstrained learning. In *International Conference on Machine Learning*, pages 4051–4060. PMLR.

David JC MacKay. 1992. The evidence framework applied to classification networks. *Neural computation*, 4(5):720–736.

Ali Malik, Volodymyr Kuleshov, Jiaming Song, Danny Nemer, Harlan Seymour, and Stefano Ermon. 2019. Calibrated model-based deep reinforcement learning. In *International Conference on Machine Learning*, pages 4314–4323. PMLR.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. 2014. Binary classifier calibration: Non-parametric approach. *arXiv preprint arXiv:1401.3390*.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2015, page 2901. NIH Public Access.

Khanh Nguyen and Brendan O'Connor. 2015. Posterior calibration and exploratory analysis for natural language processing models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1587–1598.

John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Amir Rahimi, Kartik Gupta, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. 2020a. Post-hoc calibration of neural networks. *arXiv preprint arXiv:2006.12807*.

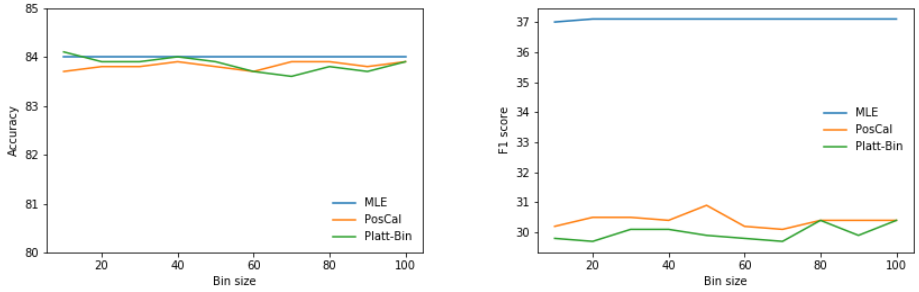
Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. 2020b. Intra order-preserving functions for calibration of multi-class neural networks. *Advances in Neural Information Processing Systems*, 33:13456–13467.

David Widmann, Fredrik Lindsten, and Dave Zachariah. 2019. Calibration tests in multi-class classification: A unifying framework. *Advances in Neural Information Processing Systems*, 32.

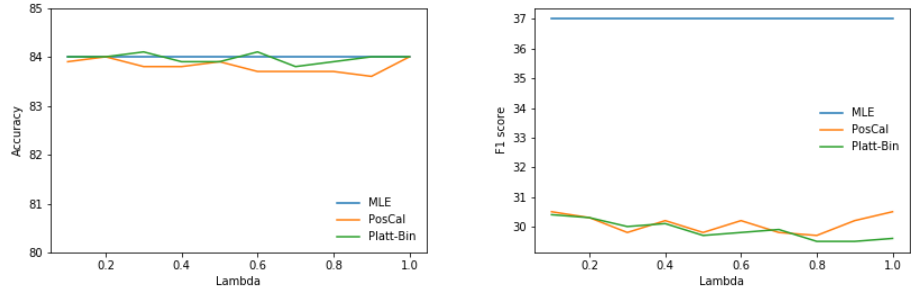
Dong Yu, Jinyu Li, and Li Deng. 2011. Calibration of confidence measures in speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2461–2473.

Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616. Citeseer.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699.



(a) Accuracy (b) F1 score



(c) Accuracy (d) F1 score

Figure 3: Effect of Bin-size (upper row) and regularization (lower row) on model accuracy and F1 score

True Label	MLE → PB	MLE $\hat{p}$	PB $\hat{p}$	Sentence
happiness	INCOR→COR	0.32	0.70	Our pleasure . Please fill out this form , leaving your address and telephone number .
noemotion	INCOR→COR	0.30	0.55	sounds good . What are you going to have for your main course ?
surprise	INCOR→COR	0.17	0.60	Really ? What did you get one for ?
happiness	INCOR→COR	0.13	0.82	I'm glad to help you . What's wrong ?
anger	INCOR→COR	0.12	0.36	Damn it ! I'm injured here . We could wait all day for the police .
anger	INCOR→COR	0.14	0.41	To hell with you . The accident was your fault .
anger	INCOR→COR	0.11	0.39	To hell with you .
noemotion	COR→INCOR	0.73	0.43	No problem .
noemotion	COR→INCOR	0.99	0.31	Of course . The fitting room is right over there .
happiness	COR→INCOR	0.61	0.46	Great , thanks .
noemotion	COR→INCOR	0.78	0.34	Hello !
happiness	COR→INCOR	0.64	0.15	Sure thing , follow me . This here is the .
noemotion	COR→INCOR	0.90	0.36	Well , if you ever want to visit Korea , I would be happy to show you around .
anger	COR→INCOR	0.41	0.36	I might just ! Enjoy your stupid game !
noemotion	COR→INCOR	0.81	0.40	But he seems to be very happy with Rose .
happiness	COR→INCOR	0.53	0.08	So sorry . Next time we'll go , thanks anyway .
disgust	COR→INCOR	0.49	0.28	I dislike it most .
noemotion	COR→INCOR	0.98	0.42	It was a real red letter day for you .
noemotion	COR→INCOR	0.96	0.37	Yeah . We rolled out the red carpet to welcome him home .

Table 4: Additonal examples for predicted $\hat{p}$ of true label from MLE and PB with corresponding sentences in DailyDialog

True Label	MLE → PB	MLE $\hat{p}$	PB $\hat{p}$	Sentence
offensive	INCOR→COR	0.02	0.56	@aschops absolutely agree with that statement. It's just so amusing how angry it makes all these teabagger scumbags. That alone is worth i
neither	INCOR→COR	0.02	0.91	@HBergHattie @snkscoyote I wonder if the progs didn't relegate young black men to the ghettos to keep them away from harry reid's friends.
offensive	INCOR→COR	0.03	0.49	kieffer_jason i swear u a fuck nigga u a scary little bitch u think this a game hu
hate	INCOR→COR	0.32	0.60	@ImToBlame you a fatherless wallet carrying ass video game playing ass negro breh. You filth. No way you can afford to date a #TwitterHone
offensive	INCOR→COR	0.09	0.74	I hate a don't get shit done ass nigg
hate	INCOR→COR	0.002	0.65	Every spic cop in #LosAngeles is loyal to the #LatinKin
offensive	COR→INCOR	0.99	0.06	"@KingCuh: @16stanleys io io alu record ho vine sai pe hahahaha" lol anywaaaaaays..... ha
hate	COR→INCOR	0.98	0.04	#RebelScience .....is using an ACTUAL WOMAN as a genetic engineering lab for "all natural clones"..... or something..... #faggot #ro
offensive	COR→INCOR	0.99	0.38	"Let's do nips ahoy and spank me mayb
hate	COR→INCOR	0.95	0.37	"Our people". Now is the time for the Aryan race 2 stand up and say "no more". Before the mongerls turn the world into a ghetto slum. 14
offensive	COR→INCOR	0.68	0.47	😒RT @SedSince81: niggers RT @VonshayeB Before any moves are made... my black ass must take a na

Table 5: Additonal examples for predicted $\hat{p}$ of true label from MLE and PB with corresponding sentences in HateOffensive

True Label	MLE → PB	MLE $\hat{p}$	PB $\hat{p}$	Sentence
impolite	INCOR→COR	0.34	0.52	What user list? The one I linked to?
polite	INCOR→COR	0.35	0.60	As I wrote above, at first I thought lets keep it, but after I heard some arguments, and when I made analysis of my own, I got to my conclusion. What's yours?
impolite	INCOR→COR	0.47	0.74	You and <url> are getting quite close to an edit war. Perhaps you should talk it out?
polite	INCOR→COR	0.16	0.63	Hey, long time no seeing! How's stuff?
polite	COR→INCOR	0.59	0.36	I am not sure of the question. Do you want problems that are obviously in one of the classes but not the other?
polite	COR→INCOR	0.62	0.45	092011 Try adding "ServerAlias mysite.com" after "ServerName" line. Also, do you have a DNS entry for mysite.com – same as www.mysite.com?
polite	COR→INCOR	0.68	0.24	I like the first shot. Are those doghouses?
impolite	COR→INCOR	0.51	0.44	Hmmm, Apple software on Windows question. I guess the "Apple Software" part defines the fact that you posted it here?
polite	COR→INCOR	0.61	0.49	how do you import the .csv into the spreadsheet? ('importdata'?)
impolite	COR→INCOR	0.68	0.48	I usually just boil water and then drink but I think it won't help here. Does it?
impolite	COR→INCOR	0.78	0.27	What's the benefit of the horizontal dropout? Is it safety? Is it just a style? Is it ease of maintenance?
impolite	COR→INCOR	0.51	0.32	Maybe it's necessary to phrase this another way: is there any food that *everybody* can eat?

Table 6: Additional examples for predicted $\hat{p}$ of true label from MLE and PB with corresponding sentences in StanfordPoliteness