Supplementary Material
Sampling Bias in Deep Active Classification: An Empirical Study

	AGN	SGN	DBP	YHA	YRP	YRF	AMZP	AZMF
#Class	4	5	14	10	2	5	2	5
#Train	120k	450k	560k	1.4M	560k	650k	3.6M	3.0M
#Test	7.6k	60k	70k	60k	38k	50k	400k	650k

Table 1: Details about the dataset sizes (both train and test) along with the number of classes.

In this document, we present statistics, additional tables and hyperparameters left out of the main work due to lack of space.

1 Dataset Details

Details of the train, test sizes and number of classes for each dataset can be found in Table 1.

2 Experiment Hyperparameters

In this section, we detail the complete list of hyperparameters, for reproducibility. We will release our code on Github.

2.1 Models

We describe the model hyperparameters used for 4 models: (i) FastText (ii) SVM (iii) ULMFiT (iv) Multinomial Naive Bayes for reproducibility.

2.1.1 FastText

We use the original implementation[1]. The hyperparameters used for each dataset can be found in Table 2. We chose to use the zipped version of FastText for optimized memory usage without loss of accuracy, or speed.

2.1.2 ULMFiT

For ULMFiT, we used the default hyperparameters from the author's implementation[2], except the batch size which we set to 32. We recall that ULMFiT has two steps: the fine-tuning of the language model and the fine-tuning of the classifier. We initialized the language model with the pre-trained weights released by the authors. Results of a pre-training on Wikitext-103 consisting of 28,595 pre-processed Wikipedia articles and 103 million words. For each compressed datasets (small and very small), we fine-tuned the language model and the classifier for 10 epochs. For fine-tuning both language model and classifier, we used a NVIDIA Tesla V100 16GB.

The hyperparameters for the language model are: batch size of 32, learning rate of 4e-3, bptt of 70, embedding size of 400, 1150 hidden units per hidden layer and 3 hidden layers. Adam Optimizer with $\beta_1 = 0.8$ and $\beta_2 = 0.99$. The dropout rates are: 0.15 between LSTM layers, 0.25 for the input layer, 0.02 for the embedding layer, 0.2 for the internal LSTM recurrent weights.

The hyperparameters for the classifier are: batch size of 32, learning rate of 0.01, embedding size of 400, 1150 hidden units per hidden layer and 3 hidden layers. Adam Optimizer with $\beta_1 = 0.8$ and $\beta_2 = 0.99$. The dropout rates are: 0.3 between LSTM layers, 0.4 for the input layer, 0.05 for the embedding layer, 0.5 for the internal LSTM recur-

Dsets	Emb Dim	NGrams	Epochs	LR	Acc Full
SGN	25	2	10	0.25	96.9
TQA	25	2	20	0.75	97.2
DBP	25	2	10	1	98.6
YHA	25	2	10	0.02	72.1
YRP	25	2	10	0.05	95.6
YRF	25	2	10	0.05	63.6
AGN	25	2	10	0.25	92.1
AMZP	25	2	10	0.01	94.2
AMZF	25	2	10	0.01	59.6

Table 2: Hyperparameters Used for FastText: Embedding dimension, Number of n-grams, number of epochs, learning rate, accuracy obtained using the full train set

Dsets	NLL	BrierL	ECE	VarR	ENT	STD
SGN	0.14	0.01	0.01	0.02	0.07	0.39
DBP	0.07	0.0	0.01	0.0	0.02	0.26
YHA	1.37	0.05	0.16	0.12	0.5	0.27
YRP	0.16	0.04	0.02	0.03	0.11	0.47
YRF	1.15	0.11	0.17	0.21	0.73	0.31
AGN	0.46	0.03	0.04	0.02	0.08	0.42
AMZP	0.26	0.05	0.04	0.02	0.08	0.48
AMZF	1.32	0.12	0.21	0.22	0.77	0.31

Table 3: Metrics measured after training FastText (FTZ-Ent) model on the resulting sample, with 39 queries, using entropy query strategy. We observe that NLL and Multiclass Brier Score remains low. The model is also well calibrated, i.e. gives calibrated uncertainty estimates.

[1] https://github.com/facebookresearch/fastText/

[2] https://github.com/fastai/fastai/tree/master/courses/dl2/imdb_scripts

Dsets	Chance	FTZ Ent-Ent	FTZ Ent-LC	MNB Ent-Ent	MNB Ent-LC	FTZ Ent-Ent	FTZ Ent-LC	MNB Ent-Ent	MNB Ent-LC
SGN	9.4 ± 0.0	81.6 ± 0.1	80.1 ± 0.3	52.9 ± 0.0	39.8 ± 0.0	74.8 ± 0.3	73.4 ± 0.6	35.0 ± 0.0	34.1 ± 0.0
DBP	9.3 ± 0.0	82.6 ± 0.2	82.2 ± 0.1	84.9 ± 0.0	69.8 ± 0.0	77.4 ± 0.1	76.6 ± 0.2	79.5 ± 0.0	64.1 ± 0.0
YHA	19.0 ± 0.0	75.0 ± 0.1	71.6 ± 0.1	90.9 ± 0.0	76.8 ± 0.0	66.1 ± 0.1	66.7 ± 0.1	86.4 ± 0.0	72.0 ± 0.0
YRP	9.3 ± 0.0	59.4 ± 0.3	59.6 ± 0.4	32.5 ± 0.0	32.5 ± 0.0	56.4 ± 0.7	56.4 ± 0.6	19.9 ± 0.0	19.9 ± 0.0
YRF	19.0 ± 0.0	75.1 ± 0.1	62.0 ± 0.1	69.6 ± 0.0	60.7 ± 0.0	67.2 ± 0.3	53.6 ± 0.1	55.5 ± 0.0	44.8 ± 0.0
AGN	19.1 ± 0.0	75.8 ± 0.3	75.1 ± 0.1	$?81.1 \pm 0.0$	71.5 ± 0.0	70.6 ± 0.2	69.1 ± 0.0	76.2 ± 0.0	67.3 ± 0.0
AMZP	9.5 ± 0.0	60.2 ± 0.1	60.2 ± 0.3	32.2 ± 0.0	32.2 ± 0.0	52.7 ± 0.6	52.7 ± 0.1	23.5 ± 0.0	23.5 ± 0.0
AMZF	19.0 ± 0.0	64.8 ± 0.3	58.5 ± 0.3	64.2 ± 0.0	57.4 ± 0.0	55.2 ± 0.1	48.4 ± 0.1	55.2 ± 0.0	50.4 ± 0.0

Table 4: Intersection across query strategies using 19 and 9 iterations (mean $\pm$ std across runs) and different seeds

Datasets	Limit	FTZ ($\cap Q$)	MNB ($\cap Q$)	FTZ ($\cap Q$)	MNB ($\cap Q$)	FTZ ($\cap Q$)	MNB ($\cap Q$)
SGN	1.6	1.5 ± 0.1	1.4 ± 0.2	1.6 ± 0.0	1.3 ± 0.2	1.6 ± 0.0	1.2 ± 0.2
DBP	2.6	2.5 ± 0.1	2.3 ± 0.1	2.5 ± 0.1	2.3 ± 0.2	2.5 ± 0.1	2.3 ± 0.1
YA	2.3	2.3 ± 0.0	2.3 ± 0.0	2.3 ± 0.0	2.2 ± 0.0	2.3 ± 0.0	2.2 ± 0.0
YRP	0.7	0.7 ± 0.0	0.7 ± 0.1	0.7 ± 0.0	0.6 ± 0.2	0.7 ± 0.0	0.7 ± 0.0
YRF	1.6	1.6 ± 0.0	1.5 ± 0.1	1.6 ± 0.0	1.4 ± 0.2	1.6 ± 0.0	1.3 ± 0.2
AGN	1.4	1.3 ± 0.0	1.3 ± 0.1	1.3 ± 0.1	1.1 ± 0.2	1.3 ± 0.0	1.1 ± 0.1
AMZP	0.7	0.7 ± 0.0	0.7 ± 0.1	0.7 ± 0.0	0.7 ± 0.0	0.7 ± 0.0	0.7 ± 0.0
AMZF	1.6	1.6 ± 0.0	1.6 ± 0.0	1.6 ± 0.0	1.6 ± 0.1	1.6 ± 0.0	1.6 ± 0.0

Table 5: Class Bias Experiments: Average Label entropy (mean $\pm$ std) across query iterations, for 39, 19 and 4 query iterations each.

rent weights.

2.1.3 Multinomial Naive Bayes (MNB)

We use the scikit-learn implementation of Multinomial Naive Bayes [3] with default hyperparameters: smoothing parameter $\alpha = 1.0$, fit prior set to True and class prior set to None. As input to our MNB, we use the scikit-learn implementation of the TFIDF Vectorizer [4]. All default hyperparameters remain unchanged except that we use a maximum feature threshold of 50000, we remove all stop words contained in the default list 'english' and we set sublinear tf to True.

2.1.4 SVM

To compute the support vectors of the datasets we used ThuderSVM, a Fast SVM library running on a V100 GPU. [5]. We used the SVC with a linear kernel, degree = 3, gamma = auto, coef0 = 0.0, C = 1.0, tol = 0.001, probability = False, classweight = None, shrinking = False, cachesize = None, verbose = False, max iter = -1, gpuid=0, maximum memory size = -1, random state = None and decision function = 'ovo'.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

[4] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[5] https://github.com/Xtra-Computing/thundersvm

3 Experiments

3.1 Class Bias

We provide in Table 5 the complete results of our class bias experiments with 39, 19 and 4 iterations using entropy query strategy.

3.1.1 Results Across Iterations

We provide in Table 4 the results of our intersection experiments for 19 and 9 iterations using entropy query strategy for FastText (FTZ) and Multinomial Naive Bayes (MNB).

3.2 Metrics Affecting Uncertainty

We provide in Table 3 several metrics measured on the resulting samples of each dataset after 39 queries and using the entropy query strategy. NLL denotes the negative log-likelihood, BrierL denotes the Brier Score Loss, ECE denotes the expected calibration error, VarR denotes the variation ratio, ENT denotes the entropy, STD denotes the standard deviation. We measure these properties of the predicted sample and compute their average over the dataset. We observe that the FastText model is well calibrated except for YRF and AMZF. Similar trends are observed in the average uncertainty measures.

3.3 Accuracy Plots for Remaining Datasets

We show in Figure 1 the accuracy curves for FastText and NaiveBayes, for 4, 9, 19 and 39 iterations using entropy query strategy vs random.

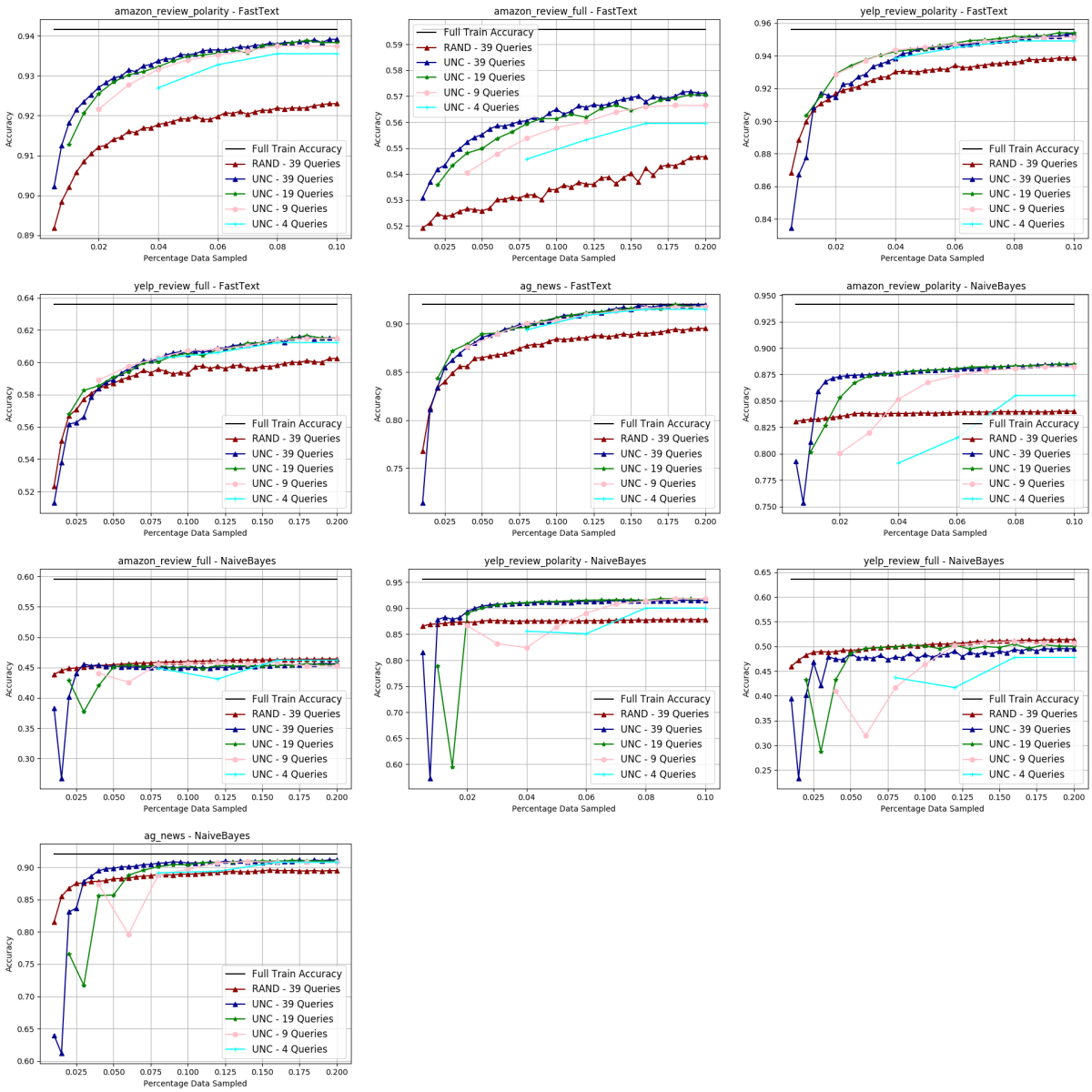

Figure 1: Accuracy across different number of queries b for FastText and Naive Bayes, with $b \times K$ constant. FastText is robust to increase in query size and significantly outperforms random in all cases