Doctoral Minimum

Motivation

The generalized aim of Natural Language Processing (NLP) is a decision making process given the contextual understanding. Over the last decade, the field made significant progress thanks to both spotlight attention of the commercial sector and many scientists and also to the immense investment and improvements in computational power. The current list of NLP tasks includes but is not limited to such problems like: topic classification, machine translation, named entity recognition, language modeling, coreference resolution, etc.. Astonishingly, a vast amount of task-specific solutions show almost human-like semantic understanding that brought even more attention to the field. The increased research activity resulted in a brain gain and a strong push of the boundaries in a majority of NLP problems, i.e representing language in a vector space. The newer and more complex models' architecture approaches, accompanied by the motto "the more parameters, the better", made the retraining more complicated and costly. However, the innovative fine tuning approaches present efficient solutions for the problem of the full retraining paradigm. The non arguable outcome of these improvements is an increase in the number of training instances i.e language models are usually trained with billions of text data.

Our research contribution lies in optimal models learning both from scratch or fine tuning of the pre-trained parameters. The training data collection is a highly time consuming and complicated procedure. In terms of supervised learning, the target labels acquisition is sometimes exceedingly expensive, e.g legal documents relevancy labeling. The idea of the paper's methodology is to generalize the optimal learning and present its functionality regarding one of the most complex problems - coreference resolution. We simultaneously explore various correlated sectors such as i) active learning, ii) models' uncertainty representation and iii) coreference resolution. The topics of *active learning* and *models uncertainty representation* are codependent. We propose a more granular study and extension of the human-machine communication. Among a huge quantity of unlabeled data the machine is expected to tell to an annotator (subject matter expert or so called oracle) which labels from the set of unlabeled data will bring forth the most information about the studied problem. Hence, the learning involves less data annotations and the model is trained with less effort. The following task is formulated as a supervised learning technique trained with the data obtained by the sequential labels querying from a human expert. This branch of research involves integration of enhanced models uncertainty measurement algorithms e.g deep ensembles, MC Dropout, SGLD or Vadam to the NLP problem for the empirical model weights distribution estimate. The additional information, given the distribution of predicted labels and the model prediction uncertainty, allows for faster model learning (hot and warm start methods) with a lower number of training data. The architecture agnostic generalization of the empirical model weights distribution estimate will grant us more freedom in NLP models selection while preserving the precise insight into the model processes given prediction-based uncertainty. The model uncertainty measurement and representation are done through the empirical estimate and sampling from the model weights distribution. The uncertainty representation approach provides the model with an expanded vision of both model learning and inference. The described technique has shown that such algorithms may enhance the learning process significantly.

The subsequent branch of this paper is the active learning and model uncertainty integration into the coreference resolution problem. Coreference resolution (CR) combines detection and linking various mentions of entities within the text: linking noun phrases with their counterparts and pronouns, anaphora disambiguation, linking words with their pro-forms, etc.. Hence, CR-solving models significantly impact the quality of the text mining algorithms. The state-of-art CR models' architectures are not as massive as the language models and not as trivial as different NLP tasks. Thus, our inclination towards more complex neural network structures, tremendous requirements for context understanding, vast usage potential, and personal interest make this type of model the perfect candidate for uncertainty algorithms integration and improvement. A deeper insight into the models' decision making process given the uncertainty is expected to allow for a new coreference resolution state-of-art threshold to be set.

An exemplary use case where coreference resolution can be applied is categorizing entities and their pronouns to provide one with a broader spectrum of information for future decision making. Based on the extracted data, it is possible to unify all knowledge in the form of a Knowledge Graph (KG) which can be further utilized for linking concepts represented by textual spans. Dependencies and connections between the entities can enrich the feature space with highly discriminative samples for other tasks. For example, assume that we have the following two consecutive sentences: "John Smith and Amanda Brown are accountants at XYZ company. Amanda's colleague was accused of drunk driving". Based on these sentences, one would wish to classify if some of the entities from the text can be charged with a misdemeanor. For a human reader, it is evident that Amanda's colleague refers to John. However, for a machine, that is a challenging task. Therefore, proper identification of entity clusters like John Smith, Amanda's colleague, Amanda Brown, XYZ would significantly improve the machine's understanding of the text. Another potential application of coreference resolution lies within the problem of opinion mining in media resources, where people frequently express their views and opinions. For example, heated discussions may emerge under political news articles. In these discussions, participants refer to subjects of the particular article with, for instance, pronouns. Therefore, proper CR may provide better traction of the audience's attitude towards entities from the article by linking comment mentions to them.

Part I
Single Instance Active Learning [1]

Supervised classification of texts relies on the availability of reliable class labels for the training data. However, the process of collecting data labels can be complex and costly. A standard procedure is to add labels sequentially by querying an annotator until reaching satisfactory performance. Active learning is a process of selecting unlabeled data records for which the knowledge of the label would bring the highest discriminability of the dataset. In this part, we provide a comparative study of various active learning strategies for different embeddings of the text on various datasets. We focus on Bayesian active learning methods that are used due to their ability to represent the uncertainty of the classification procedure. We compare three types of uncertainty representation: i) SGLD, ii) Dropout, and iii) deep ensembles. The latter two methods in cold- and warm-start versions. The texts were embedded using Fast Text, LASER, and RoBERTa encoding techniques. The methods are tested on two types of datasets, text categorization (Kaggle News Category and Twitter Sentiment140 dataset) and fake news detection (Kaggle Fake News and Fake News Detection datasets). We show that the conventional dropout Monte Carlo approach provides good results for the majority of the tasks. The ensemble methods provide more accurate representation of uncertainty that allows to keep the pace of learning of a complicated problem for the growing number of requests, outperforming the dropout in the long run. However, for the majority of the datasets the active strategy using Dropout MC and Deep Ensembles achieved almost perfect performance even for a very low number of requests. The best results were obtained for the most recent embeddings RoBERTa

1 Introduction

The development of a text classifier on a new problem requires the availability of the training data and their labels. Labeling involves human annotators and a common practice is to label as many text documents as possible, train a classifier and search for new data and labels if the performance is unsatisfactory. Random choice of the documents for the data set extension can be costly because the new documents may not bring new information for the classification. Active learning strategy aims to select among available unlabeled documents those that the classifier is most uncertain about and queries an annotator for their labels. Therefore, it has the potential to greatly reduce the effort needed for the development of a new system. While it was introduced almost two decades ago, recent improvements in deep learning motivate our attempt to revisit the topic. For example, SVM-based active learning approaches for text classification date back to 2001 [33], where the superiority of active learning over random sampling was demonstrated. Since deep recurrent and convolutional neural networks achieve better classification results, Bayesian active learning methods for deep networks gained popularity especially in image classification [13], [22].

The Bayesian approach is concerned with querying labels for data for which the classifier predicts the greatest uncertainty. The uncertainty is quantified using the so-called acquisition function, such as predictive variance or predictive entropy [29]. While different acquisition functions often provide similar results, different representations of predictive distribution yield much more diverse results. The most popular approach using Dropout MC [13] has been tested on text classification [3] and named entity recognition [30], [22], however other techniques

[1] The major part of the content in this part was taken and restructured from [27].

such as Langevin dynamics [37] and deep ensembles [20] are available. Deep ensembles often achieve better performance [5], [32] but require higher computational cost since they train an ensemble of networks after each extension of the data set. One potential solution of this problem has been recently proposed in [34], where the ensemble is not trained from a fresh random initialization after each query but initialized randomly around the position of the ensembles from the previous iteration. In this contribution, we test this approach and compare it with the dropout MC and Langevin dynamics representations. We also provide sensitivity study for the choice of the hyperparameters. Active learning for fake news detection was considered in [6] using uncertainty based on probability of classification. It was later extended to a context aware approach [7]. An entropy based approach has been presented in [15] using an ensemble of three different kinds of networks.

2 Methods

Throughout the paper, we will use three different embedding algorithms such as Fast Text [24], LASER [4]and RoBERTa [21]. RoBERTa is a modified BERT transformer model [10] based on multi-head attention layers [35]. Transformer models provide state of the art results in context understanding and it is advantageous to compare behavior of active learning algorithms with respect to different embedding techniques. Representation of the $i-$th text document $\mathbf{x}_i$ is calculated as the mean value from sentence embeddings of all sentences in the text

$$\mathbf{x}_i = \frac{1}{|D_i|} \sum_{j \in D_i} f_{\text{sentence embed}}(C^{(j)})$$

where D_i is the set of vectors where each vector represents a sentence in the $i-$th document, $|D_i|$ is a cardinality of D_i , $C^{(j)}$ is a matrix of $j-$th sentence where words are encoded with one hot or byte pair encoding [31] technique and $f_{\text{sentence embed}}$ is a function that creates sentence embeddings with respect to the given one-hot or byte pair encoded words. Fast Text encoding is made with respect to one-hot encoded words. LASER and transformer based models take byte-per encoded words as an input. Sentence embeddings for LASER and RoBERTa are output of deep neural network models. Sentence embedding for the Fast Text model is calculated as a mean value of all embeddings of words in the sentence. For supervised classification, each document embedding $\mathbf{x}_i$ has to have an associated label $\mathbf{y}_i$. We are concerned with binary classification for simplicity, however, an extension to multiclass is straightforward. We assume that for the full corpus of text documents $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]$, only an initial set of $l_0 \ll n$ labels is available, $\mathbf{Y}^{(0)} = [\mathbf{y}_0, ..., \mathbf{y}_{l_9}]$, splitting the full set $\mathbf{X}$ to the labeled, $\mathbf{X}^{(0)} = [\mathbf{x}_1, ..., x_{l_0}]$, and unlabeled parts, $\mathbf{X} \backslash \mathbf{X}^{(0)}$. Active learning is defined as a sequential extension of the training data set. In each iteration, $l = 1, ..., L$, the algorithm computes entropy of the predictive probability distribution for each document in the unlabeled dataset and selects the index of the document with the highest entropy (entropy acquisition function), formally:

$$k_l = \arg\max_{k \in K} \mathbb{E}_{p(\theta|\mathbf{X}^{(l-1)}, \mathbf{Y}^{(l-1)})} \Big(H(\mathbf{y}_k|\theta) \Big) \tag{1}$$

$$H(\mathbf{y}_k|\theta) = \mathbb{E}_{p(\mathbf{y}_k|\theta, \mathbf{x}_k)}(-\log p(\mathbf{y}_k|\theta, \mathbf{x}_k) \tag{2}$$

where K is the set of indexes of all unlabeled documents, $\mathbb{E}$ is the expectation operator over the posterior probability of the classifier parameters θ trained on all labeled data $p(\theta|\mathbf{X}^{(l-1)}, \mathbf{Y}^{(l-1)})$ and $H(\mathbf{y}_k|\theta)$ is conditional entropy of the predicted class for $\mathbf{x}_k$. The document of the selected index is sent to the human annotator with a request for labeling. When the selected text is annotated, the text is added with its label to the labeled data set $\mathbf{X}^{(l)} = [\mathbf{X}^{(l-1)}, \mathbf{x}_{k_l}]$, $\mathbf{Y}^{(l)} = [\mathbf{Y}^{(l-1)}, \mathbf{y}_{k_l}]$. The procedure is repeated L times. The active learning process is visualized in Figure 1.

The key component of the method is a representation of the posterior distribution of the parameter θ. Due to the complexity of the neural networks it is always represented by samples, with a different method of their generation. We will compare the following methods: i) SGLD: Stochastic Gradient with Langevin dynamics [37], which adds additional noise to the gradient in stochastic gradient descent, ii) Dropout MC: is an extension of the ordinary dropout that samples binary mask multiplying output of a layer, hence stopping propagation through all neurons where zeros is sampled through the network. The extension applies the sampled mask even for predictions generating samples from the predictive distribution [13], iii) Deep ensembles: consist of N networks trained in parallel from different initial conditions [20], and iv) Softmax uncertainty: is the most simple approach that uses only one network to estimate a single value θ and maximizing entropy $H(\mathbf{y}_k|\theta)$ instead of the expectation 1 [6]. Ensembles based approach is the current state-of-the-art in active learning [5]. While many of these have been tested in active learning, the majority of authors assumed that after each step of active learning, the network training starts from the initial conditions. This is clearly suboptimal, since the information from the previous training is lost. A simple solution was presented in [34], where it was argued that estimated results from

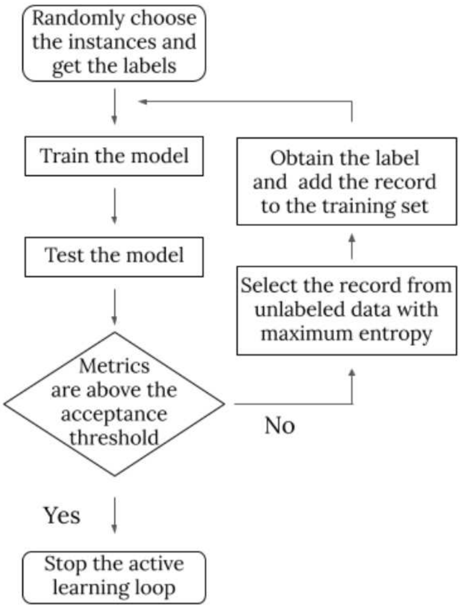

Randomly choose the instances and get the labels
Train the model
Test the model
Metrics are above the acceptance threshold
Yes
Stop the active learning loop
No
Select the record from unlabeled data with maximum entropy
Obtain the label and add the record to the training set

Figure 1: Flowchart of the active learning algorithm

the previous step can be used as centroids around which the new initial point is sampled. Since this is a form of a warm-start, we also test warm-start strategies for Dropout. The methods for representation of parametric uncertainty are:

Deep Ensemble Filter (DEnFi):

is a deep ensemble method with 10 neural networks in the ensembles and warm-start training strategy [20] using weights of the ensemble members in the previous iteration as initial conditions for the new ensemble. Each weight is perturbed by an additive Gaussian noise of variance q which is a hyperparameter. In our experiments, the ensemble is trained with parameters initialization_epochs = 2000 on the initial data and with additional warm_start_epochs = 700 epochs after each extension of the learning data set. DEnFi algorithms are displayed in Algorithms 1.

Algorithm 1 DEnFi active learning training algorithm

```
Initialize: ensemble classifier structure $\mathbf{y} = \mathbf{y}(\mathbf{x}, \theta_1^{(0)}, ..., \theta_N^{(0)})$, where $\theta_j^{(0)},\ j \in \{1, ..., N\}$ are different initializations of classifier parameters. Initial data $\mathbf{Y}^{(0)}, \mathbf{X}^{(0)}$. Noise pertrubation variance is $q$. Set iteration counter $i = 0$
Iterate until a stopping condition:
    1. For each parameters' set $\theta_j^{(i)},\ j \in \{1, ..., N\}$ provide
        (a) noise pertrubation as $\theta_j^{(i)} = \theta_j^{(i)} + \gamma_j^{(i)},\ \gamma_j^{(i)} \sim \mathcal{N}(0, q\mathrm{I})$
        (b) Train a classifier for $\mathbf{X}^{(i)}, \mathbf{Y}^{(i)}$ and initial number of epochs $\gg$ warm start epochs if $i = 0$, else warm start epochs
    2. Sample a new instance from Equation 1 and update $\mathbf{X}^{(i+1)} = [\mathbf{X}^{(l)}, \mathbf{x}_{k_{i+1}}]$, $\mathbf{Y}^{(i+1)} = [\mathbf{Y}^{(i)}, \mathbf{y}_{k_{i+1}}]$, $i = i + 1$
```

Dropout MC:

is the standard algorithm [13] that trains only a single network with sampled dropout indices and uses the sampling even in the prediction step. Generation of the Monte Carlo prediction is obtained by sampling different values of the dropout binary variable and one forward pass of the network for each sample. We study three versions of the algorithm: i) cold-start with 3000 epochs after each request, ii) warm-start with weights from the previous iteration perturbed by an additive noise of variance q with 700 epochs and iii) hot-start, with 50 epochs after each request without perturbation (hot start is a warm-start method with $q = 0$). Dropout rate is 0.5.

SGLD:

variance of the noise added to the gradient descent:

$$\hat{\theta}_{n+1} = \hat{\theta}_n - \frac{|\mathbf{X}^{(l)}|}{|\mathbf{X}^{(l)}|_{\text{minibatch}}} \frac{\eta_n}{2}\Big(\nabla L\big(\mathbf{X}^{(l)}, \mathbf{Y}^{(l)}, \hat{\theta}_n\big)\Big) + \epsilon_n,\ \epsilon_n \sim \mathcal{N}(0, \eta_n \mathbf{I})$$

where L is a loss function and ϵn is noise with covariate matrix $\eta_n\mathbf{I}$ where η_n is the learning rate of the SGD algorithm. We choose initial value of η_1 to be 0.01 and update rule $\eta_{n+1} = \frac{\eta_n}{b-3000} + 0.05$. The noise ϵ_n is added to the gradient only after 3000 of initial training epochs. Then we draw 50 samples with 100 epochs between consecutive samples to avoid correlation.

Softmax uncertainty:

The simplest approach to uncertainty representation is a single neural network with a softmax output layer that considers uncertainty as the output of the softmax score. We add it to comparison since it has been applied to active learning in [6]. The model is trained to run 2000 epochs on the initial data with additional 200 epochs after each extension of the learning data set.

3 Experiments

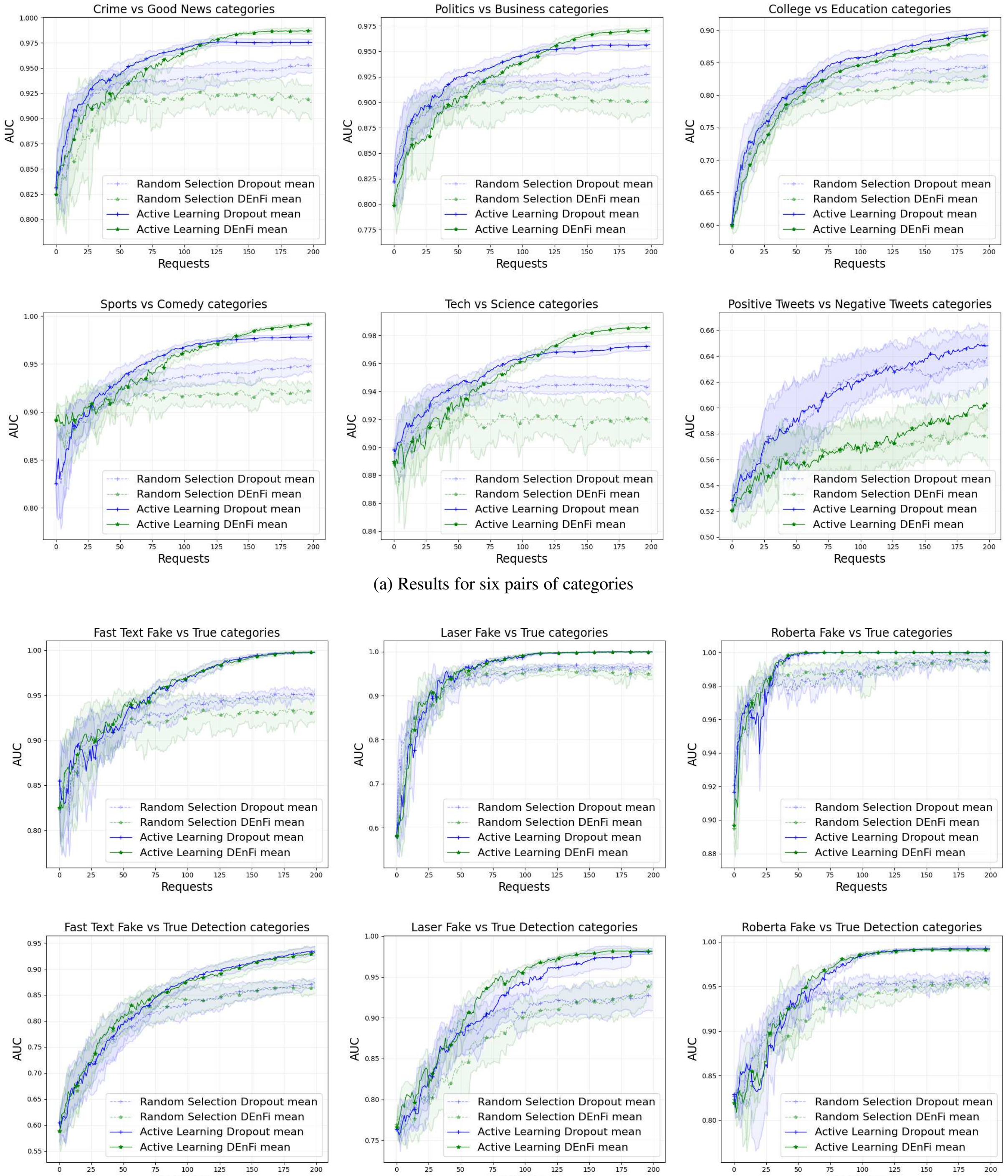
The methods were compared on the positive/negative tweets from the Tweets Dataset [14], 5 pairs of categories from the News Category Dataset [25] and two types of the news datasets for fake news detection [1], [2]. Documents from the News Category dataset were downloaded using links provided in the dataset. The names of the tested categories are shown in table 2, Figure 2a and in Figure 2a.

Specifically, we compared active learning and random sampling strategies for different settings of document embeddings and different representations of uncertainty. Each experiment was initiated by random choice of the initial training set of $l_0 = 10$ samples from 1000 text documents (500 text documents per category), which were the initial 1000 documents of the datasets. For each experiment $L = 200$ requests for annotation are simulated. The document selection follows the $\epsilon-$greedy approach [36], i.e. the sample with maximum acquisition function 1 is accepted with probability $\epsilon = \frac{\exp(l-40)}{\exp(l-40)+1}$. A random document is selected for labeling if not accepted. After each request, the classification performance is evaluated on the remaining part of the selected dataset (i.e. on the 990 text documents in the first evaluation) using the area under the ROC curve (AUC) metrics [12]. In order to make the results statistically valid, we repeat the described simulation loop 10 times for Twitter and News Category datasets and 5 times for Fake News and Fake News Detection datasets.

3.1 Hyperparameter Tuning

The classification network was designed as a feed-forward NN with one dense layer of 100 neurons with sigmoid activation functions, and softmax output layer. Warm start versions of both Dropout MC and DEnFi have hyperparameter q that governs the perturbation of the previous result before training on the extended dataset. Tuning of this hyperparameter was performed by a grid search for both the Fast Text and RoBERTa text encoding techniques. Since the main focus of the paper is on the effect of the warm-start strategy, the results of the effect of the noise variance q for DEnFi and Dropout MC is displayed in Table 1a and Table 1f for active learning strategy on the Tech vs Science task and Fake News Detection dataset, respectively.

Based on Table 1a it is clearly seen that the variance of the perturbation noise related to the best result is increasing with the number of requests. We conjecture that the variance has the role of a selector of the exploration/exploitation tradeoff. Low variance favors exploitation and improves quickly, higher variance implies less accurate guesses in the initial iterations but better performance in the long run. The results from Table 1a indicate that higher values of q are not performing well thus the range of q was reduced for tuning of the hyperparameter for the fake news datasets. The increase of the optimum value of the noise variance with the number of requests is also visible in Table 1f, especially for the RoBERTa encoded data. Since calibration of the variance for all



(a) Results for six pairs of categories

(b) Results for two datasets: Fake News (top row) and Fake News Detection (bottom row), and three embeddings: Fats Text (left column), Laser (middle column) and RoBERTa (right column)

Figure 2: Evolution of the mean AUC with growing number of requests for DEnFi and Dropout MC warm-start algorithms. The uncertainty bounds are illustrated as one standard deviation from the mean value with respect to 10 runs for top figure and 5 runs for bottom figure. Both DEnFi and Dropout MC were initially trained on 10 labeled text documents before sequential learning strategies were initialized

Noise	AUC after # of iterations			
Variance	50	100	150	200
0.1	**0.945***	**0.968***	0.974	0.976
0.2	0.932	0.964	0.976	0.978
0.3	0.930	0.961	**0.982***	0.986
0.4	0.909	0.948	0.976	**0.990***
0.6	0.874	0.921	0.952	0.979
1	0.805	0.871	0.906	0.941

(b) **Fast Text** DEnFi

Noise	AUC after # of iterations			
Variance	50	100	150	200
0.1	**0.936**	**0.966***	0.972	0.976
0.2	**0.938***	**0.966***	**0.981***	0.983
0.3	0.920	0.956	**0.982**	**0.989***
0.4	0.917	0.955	0.976	**0.988**
0.6	0.894	0.948	0.972	**0.986**
1	0.859	0.914	0.941	0.970

(c) **Fast Text** Dropout

Noise	AUC after # of iterations			
Variance	50	100	150	200
0.1	**0.935**	**0.970***	**0.982**	0.988
0.2	**0.940***	0.954	**0.983***	**0.990***
0.3	0.903	0.930	0.967	0.987
0.4	0.883	0.911	0.945	0.976
0.6	0.799	0.853	0.885	0.945
1	0.661	0.750	0.818	0.875

(d) **RoBERTa** DEnFi

Noise	AUC after # of iterations			
Variance	50	100	150	200
0.1	**0.930***	**0.970**	0.982	0.986
0.2	**0.923**	**0.971***	**0.986***	**0.990***
0.3	0.917	0.959	0.982	**0.990***
0.4	0.878	0.949	0.974	**0.990***
0.6	0.822	0.908	0.952	0.980
1	0.643	0.741	0.862	0.921

(e) **RoBERTa** Dropout

(e) Results for Tech vs Science categories

Noise	AUC after # of iterations			
Variance	50	100	150	200
0.1	**0.794**	**0.886***	**0.910**	**0.942***
0.2	**0.806***	**0.877**	**0.911***	**0.932**
0.3	**0.806***	0.866	**0.901**	**0.911**

(g) **Fast Text** DEnFi

Noise	AUC after # of iterations			
Variance	50	100	150	200
0.1	**0.806**	**0.879***	**0.916***	**0.949***
0.2	**0.804**	**0.884***	0.914	0.939
0.3	**0.809***	**0.878**	**0.912**	**0.944**

(h) **Fast Text** Dropout

Noise	AUC after # of iterations			
Variance	50	100	150	200
0.1	**0.929***	**0.986***	**0.992**	0.991
0.2	**0.923**	0.975	0.990	**0.998***
0.3	**0.891**	0.935	0.955	0.975

(i) **RoBERTa** DEnFi

Noise	AUC after # of iterations			
Variance	50	100	150	200
0.1	**0.941***	**0.984***	**0.992***	**0.993**
0.2	0.917	0.974	**0.992***	**0.995***
0.3	0.917	0.959	0.982	0.990

(j) **RoBERTa** Dropout

(j) Results for Fake and True news categories (Fake News Detection dataset)

Table 1: **Fast Text** and **RoBERTa** encoding based AUC of text classification after selected number of requests of the active learning using DEnFi and Dropout MC warm-start for various selection of the perturbation noise q. Average over 10 runs for top tables and 5 runs for bottom tables. The best result is denoted by a star, results within one standard deviation of the winner are displayed in bold font.

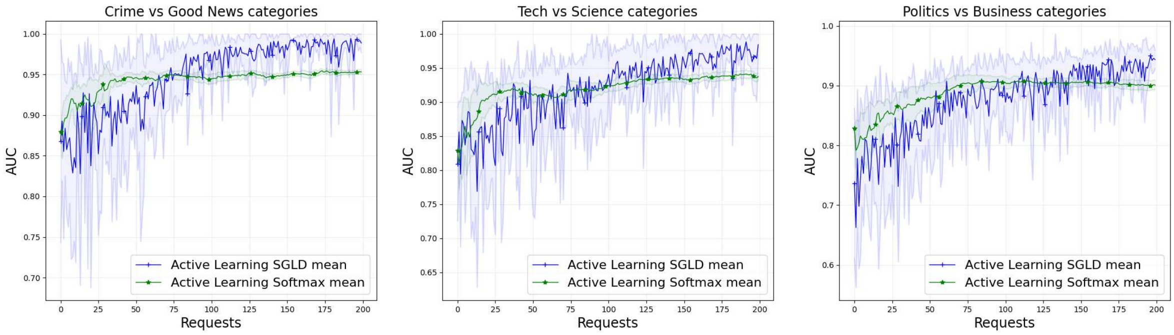


Figure 3: Evolution of the mean AUC with growing number of requests for SGLD, Softmax Uncertainty algorithms, and three pairs of categories. The uncertainty bounds are illustrated as one standard deviation from the mean value with respect to 10 runs. Both SGLD and Softmax Uncertainty were initially trained on 10 labeled text documents before sequential learning strategies were initialized

Method	Crime/ Good News	Sports/ Comedy	Politics/ Business	Tech/ Science	Education/ College	Pos./Neg. Tweets
SGLD	**0.989***	0.968	0.944	0.984	0.881	0.621
DEnFi, $q = 0.3$	0.987	**0.992***	**0.971***	**0.986**	**0.893**	0.603
MC Dropout cold-start	0.975	0.978	0.957	0.972	**0.898***	**0.648**
MC Dropout warm-start, $q = 0.3$	0.978	0.979	0.954	0.973	0.877	**0.657***
MC Dropout warm-start, $q = 0$	0.978	0.951	0.944	**0.989***	0.824	0.561
Softmax Uncertainty	0.953	0.939	0.901	0.938	0.800	0.609

Table 2: Average AUC over 10 runs for five different algorithms after 200 iterations of active learning and six different datasets. The best result is denoted by a star, results within one standard deviation of the winner are displayed in bold font.

methods and all datasets would be too computationally expensive, we ran all remaining experiments with $q = 0.3$ for Fast Text encoding and $q = 0.1$ for RoBERTa and LASER encodings. However, tuning of this hyperparameter for an application scenario or an adaptive tuning strategy offers clearly a potential for further improvement.

3.2 Text Category Classification

A comparison of AUC of the active learning strategy after 200 requests for all tested algorithms and Fast Text encoding with respect to Tweets and News Category datasets is reported in Table 2. This experiment was designed for comparison of various uncertainty representations. While all methods except for the Softmax Uncertainty achieved best results on some datasets, the most consistent results were delivered by the DEnFi and Dropout MC methods. The SGLD performance results exhibited the highest variance within the 10 runs of the method. The results for three pair categories and an active learning strategy are visualized in figure 4. Since it is the least reliable method, we will not study it any further. It is clear from Table 2 and Figure 4 3 that the simple Softmax Uncertainty provides the worst results. Since Dropout MC has comparable computational complexity, the use of simple Softmax Uncertainty is always suboptimal.

Detailed analysis of the DEnFi and Dropout MC strategies is provided in figure 2a for Tweets and News Category datasets. For better insight, we also display the performance of the random sampling learning strategy where training data are extended using randomly sampled documents. Note that for random sampling, Dropout MC outperforms consistently DEnFi on all tasks. We conjecture that this is due to the robustness of the dropout regularization. However, the power of DEnFi becomes apparent with the increasing number of requests. It is improving slower than Dropout MC at the beginning, but improves faster, thus outperforming Dropout MC in the long run. We conjecture that this is due to better exploration capability of the DEnFi while Dropout MC excels at exploitation. The speed of improvement depends on the complexity of the learning task. For simpler tasks (such as crime vs. Good News), AUC over 0.98 is achieved quickly. However, for more complex tasks, such as Positive vs Negative Tweets, the number of data needed for improvement is much higher. The active learning is on par with the random sampling up to 125 requests and even after 200 requests, the AUC is below 0.7 indicating poor performance. Note that the active learning strategy of DEnFi starts improving over the random sampling sooner than Dropout MC with a sharper slope which indicates a high probability of obtaining the same profile as the other datasets in the long run.

3.3 Fake News Classification

The fake news data experiment is made under the same initial conditions as in Section 3.2 but for a wider choice of the document encoding techniques. Based on results from Section 3.2 we show comparison of only the DEnFi and Dropout warm-start for three different types of encodings (Fast Text, LASER, RoBERTa) on the two fake news datasets in Figure 2b. The results for fake news datasets are similar to those for the Twitter and News Category datasets. Advantage of the DEnFi approach is significant only for the Fake News Detection dataset with Laser embeddings. For other cases the difference is negligible. Under the exploration/exploitation hypothesis, this means that the data sets are well separable and more sophisticated exploration capabilities of DEnFi are not needed.

The best performance of the active learning strategy was achieved for the RoBERTa embedding. Note that the embedding has much greater influence on the speed of learning than uncertainty representation.

4 Conclusion

We have studied the suitability of various uncertainty representations for the task of active text classification. The established Dropout MC methodology was compared against deep ensembles, SGLD and simple softmax strategy. To reduce the computational cost, we studied warm-start strategies for both ensembles (called DEnFi) and Dropout MC, that resulted in significant reduction of training epochs per the active learning iteration. The resulting methods exhibit a different tradeoff between exploration and exploitation. While Dropout MC has been found to be more reliable in random sampling strategy and improving faster at the beginning of active learning, the DEnFi was found to prefer exploration sacrificing performance at the beginning but outperforming Dropout MC in the longer run of active learning. The highest deviation of the random strategies between DEnFi and Dropout MC resulted in 0.04 AUC in favor of Dropout MC. However, for the active learning strategy both Dropout MC and DEnFi converged to the same numbers.

Sensitivity of the active learning to the choice of embedding was evaluated on the fake news datasets. It was observed that the most recent embedding method (RoBERTa) facilitated the fastest learning and that the choice of embedding was more important than the uncertainty representation. However, methods of active learning achieved significantly faster learning than the random sampling approach (the difference between the random and the active learning strategies differs from 0.015 AUC up to 0.080 AUC) on all tested datasets of various complexity.

Part II
Batch Active Learning

Supervised learning of classifiers for text classification and sentiment analysis relies on the availability of labels that may be either difficult or expensive to obtain. A standard procedure is to add labels to the training dataset sequentially by querying an annotator until the model reaches a satisfactory performance. Active learning is a process that optimizes unlabeled data records selection for which the knowledge of the label would bring the highest discriminability of the dataset. Batch active learning is a generalization of a single instance active learning by selecting a batch of documents for labeling. This task is much more demanding because plenty of different factors come into consideration (i. e. batch size, batch evaluation, etc.). In this paper, we provide a large scale study by decomposing the existing algorithms into building blocks and systematically comparing meaningful combinations of these blocks with a subsequent evaluation on different text datasets. While each block is known (warm start weights initialization, Dropout MC, entropy sampling, etc.), many of their combinations like Bayesian strategies with agglomerative clustering are first proposed in our paper with excellent performance. Particularly, our extension of the warm start method to batch active learning is among the top performing strategies on all datasets. We studied the effect of this proposal comparing the outcomes of varying distinct factors of an active learning algorithm. Some of these factors include initialization of the algorithm, uncertainty representation, acquisition function, and batch selection strategy. Further, various combinations of these are tested on selected NLP problems with documents encoded using RoBERTa embeddings. Datasets cover context integrity (Gibberish Wackerow), fake news detection (Kaggle Fake News Detection), categorization of short texts by emotional context

(Twitter Sentiment140), and sentiment classification (Amazon Reviews). Ultimately, we show that each of the active learning factors has advantages for certain datasets or experimental settings.

1 Introduction

Supervised learning of classifiers relies on the availability of class labels which often involves a human annotator for a majority of NLP tasks. This can be costly for large datasets. Active learning is a strategy designed to minimize this cost by automatic selection of those unlabeled documents that are expected to bring useful information for the classifier. Advantages of this approach have been demonstrated even for classical methods such as SVM [33]. The most conventional active learning strategies select only one unlabeled document after each training round to query due to the simplicity of its selection. The next query document is selected only after the first one is labeled and the model retrained, which means that the annotator has to wait for retraining. This impractical strategy can be avoided if the active learning algorithm selects a batch of documents. Novel methods for batch active learning appear frequently, each demonstrating advantages on their benchmark data.

Various comparative studies have been performed recently with various focus and results. Batch active learning was studied in [11] for only one size of the batch (50 documents per query). Large sensitivity to the type of dataset was reported in [26], where different methods won for different data. Large variability of the results was also observed in [18]. In [28], the min-margin strategy was shown to be competitive with the prediction entropy-based method on a range of embeddings. The comparative studies shared similar properties, such as a fixed network for embeddings (improvement with retraining can be expected [23] but may be too costly). All studies also assume a cold start, i.e. completely new initialization of the classifier after each round of querying. This is motivated by the fear of overfitting, which was demonstrated in [17] for hot start, i.e. continuation of training of the classifier. A compromise in the form of warm-start, i.e. adding noise to the weights of the previous classifier, was proposed in [27].

In this contribution, we take a different approach to benchmarking of the batch active learning algorithms. Specifically, we decompose the algorithms into their building blocks: i) the size of the minibatch, ii) acquisition function, iii) representation of uncertainty of the classifier, and iii) initialization of the network. This approach allows us to quantify contribution of each of the building-block and combine them in previously untested versions. This allows us to demonstrate the following contribution:

1. We present an extension of the Hierarchical Agglomerative Clustering (HAC) approach [8] to the Bayesian setting by replacing the min-margin with a Bayesian acquisition function, such as BALD [16]. This is a novel combination that has not been tested before.

2. We show that performance of various methods is clustered based on particular building blocks of the method. Thus indicating that active learning methods may be tailored for each target application. A good example of Bayesian methods lies in estimating the distribution of neural networks which performed the best on Fake news but showed the same results on other datasets.

3. In a large scale study we demonstrate that warm start is often beneficial and even simple methods (such as single neural network with entropy acquisition) provide results competitive to, or better than, complex active learning schemes. We also show that cold start approaches reach the same or even worse results than the aforementioned warm start techniques. This is encouraging for practitioners that are interested in the methodology.

The paper is organized as follows. In Section 2, we briefly review all tested factors of batch active learning. The experimental setup of the sensitivity study is described in Section 3 and the results are reported in Section 4.

2 Batch Active Learning Methods

Throughout the paper, we will use the RoBERTa embedding [21] to represent documents in the feature space. RoBERTa is a modified BERT transformer model [10] that achieved comparable performance to BERT in [28] and outperformed all other embeddings in [27]. Representation of the k-th text document $\mathbf{x}_k$ is calculated as the mean value from sentence embeddings of all sentences in the text.

The aim of document classification is to find a classifier $\hat{\mathbf{y}} = \mathbf{y}(\theta, \mathbf{x})$ predicting the class label for each document representation $\mathbf{x}$. In a supervised setting, the classifier parameters are found on a training set $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^{K}$ by matching the prediction $\mathbf{y}(\theta, \mathbf{x}_k)$ with the provided label $\mathbf{y}_k$ for each document. We are concerned with binary classification for simplicity, however, an extension to multiclass is straightforward.

We assume that for the full corpus of text documents $\mathbf{X}$, only a small initial set of labels $\mathbf{Y}^{(0)}$, is available. The full set $\mathbf{X}$ is thus split into the labeled, $\mathbf{X}^{(0)}$, and unlabeled parts, $\mathbf{X}_u^{(0)} = \mathbf{X} \backslash \mathbf{X}^{(0)}$, the training set in the first round is then $\mathbf{X}^{(0)}, \mathbf{Y}^{(0)}$. Active learning is defined as a sequential extension of the training data set following a simple iterative strategy in Algorithm 2.

```
**Algorithm 2** General batch active learning
**Initialize**: set classifier structure $\mathbf{y} = \mathbf{y}(\mathbf{x}, \theta)$, iteration counter $i = 0$, initial data $\mathbf{Y}^{(0)}, \mathbf{X}^{(0)}, \mathbf{X}_u^{(0)}$
**Iterate** until a stopping condition:
    1. Train a classifier parameter $\theta^{(i)}$ on $\mathbf{Y}^{(i)}, \mathbf{X}^{(i)}$, starting from $\theta_{\text{init}}^{(i)}$
    2. Compute the value of a label for all documents in the unlabeled dataset, $a_l = A(\mathbf{x}_l, \theta^{(i)})$,$\forall \mathbf{x}_l \in \mathbf{X}_u^{(i)}$
    3. Select a batch of documents, $\tilde{\mathbf{X}} \subset \mathbf{X}$, for labeling using $a_l$
    4. Query labels $\tilde{\mathbf{y}}$ for $\tilde{\mathbf{X}}$ and extend the training set $\mathbf{X}^{(i+1)} = \mathbf{X}^{(i)} \cup \tilde{\mathbf{X}}, \mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} \cup \tilde{\mathbf{y}}$,$i = i + 1$.
```

The general algorithm can be specialized to many variants depending on various factors as specified next. We will introduce several choices labeled by the step in which they appear in Algorithm 2.

1a. Uncertainty representation:

The uncertainty can be represented by a maximum likelihood estimate, represented by a single network, or a Bayesian probabilistic estimate, represented typically by an ensemble of networks. We will consider the following options: **Single network** with a softmax output layer predicting the normalized probability of each class in one hot encoding. This probability is conditioned on the parameter, and thus captures only aleatoric uncertainty. Uncertainty in parameters is not represented. **Ensemble of networks**, represent uncertainty in parameters by different parameter value in each ensemble thus capturing both aleatoric and epistemic uncertainty. We consider two methods for generating the ensemble members: i) *MC dropout* [13], where ensemble members are generated by random draws of the dropout layers, and ii) *deep ensembles* [20] where ensembles are trained independently. Note that MC dropout is computationally much cheaper.

1b. Initialization of the training:

Each training in step 1 is a new task. However, the data set typically overlaps with the one from the previous iteration, which motivates the following strategies of reusing results from the previous iteration. The **Cold start** strategy is not reusing any information, the networks are initialized by random numbers, $\theta_{\text{init}}^{(i)} = \mathcal{N}(0, \sigma)$, where σ is given by the standard network init strategy, used most often [11, 8, 28]. The **Hot start** strategy reuses all information, setting the estimate from the previous iteration as a starting point, $\theta_{\text{init}}^{(i)} = \theta^{(i-1)}$,criticized in [17]. The **Warm start** strategy a combination of the above, $\theta_{\text{init}}^{(i)} = \theta^{(i-1)} + \mathcal{N}(0, \sigma)$, where σ is a hyper-parameter [27].

2. Acquisition function:

Is a measure of the expected utility of the knowledge label, $\boldsymbol{y}_l$, for each document, $\boldsymbol{x}_l$, in the unlabeled data set. Different running index l is used to indicate that we operate on the unlabeled set. While many different utilities are proposed, we will study only the most popular ones. **Entropy** exists in two forms, entropy of the prediction $a_l = \mathbb{H}(y|\mathbf{x}_l, \theta)$ for a single network, or expected entropy $a_l = \mathbb{E}_\theta \mathbb{H}(y|\mathbf{x}_l, \theta)$ for ensembles. **BALD** is a mutual information metric, $a_l = \mathbb{E}_\theta \mathbb{H}(y|\mathbf{x}_l, \theta) - \mathbb{H}(y|\mathbf{x}_l)$ that is meaningful only for the ensembles. **Min-margin** is a minimum difference between class predictions $a_l = -\min_{c,d \in [1,C]}(y_c - y_d)$, where C is the number of classes. Note carefully that the extreme of this criteria is equivalent to maximum entropy for binary classification with a single network.

3. Batch selection strategy:

When only one sample is to be selected, it is optimal to choose the one with maximum utility given by the acquisition function. However, the complexity of the maximum utility grows exponentially when the strategy has to select a batch of b documents for off-line labeling. Strategies that try to approximate this selection using greedy search [19] are still too computationally expensive for large batches. Therefore, we select two batch selection

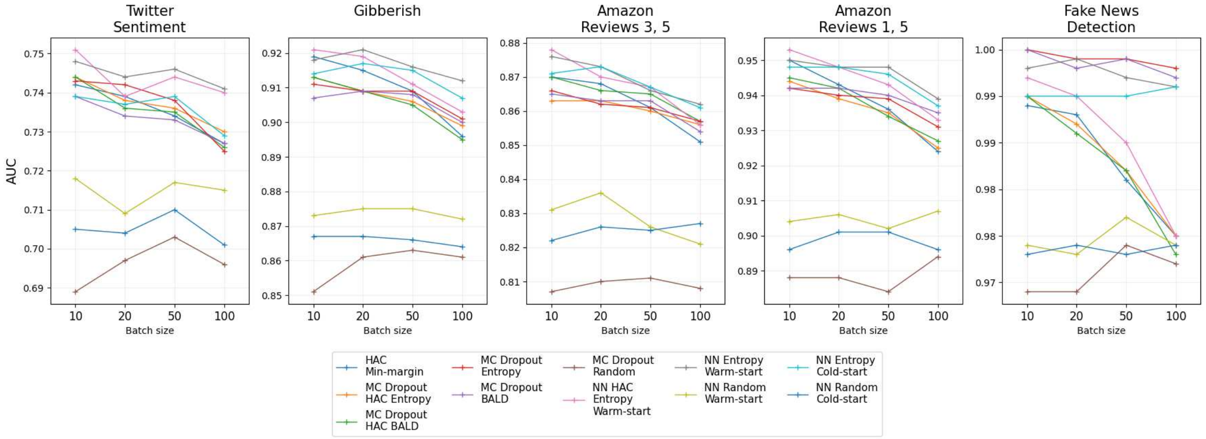


Figure 4: AUC metrics for seven active learning and two random strategies after 1000 acquired samples given datasets and batch size. Prediction of the MC dropout classifiers is an average over ensemble members.

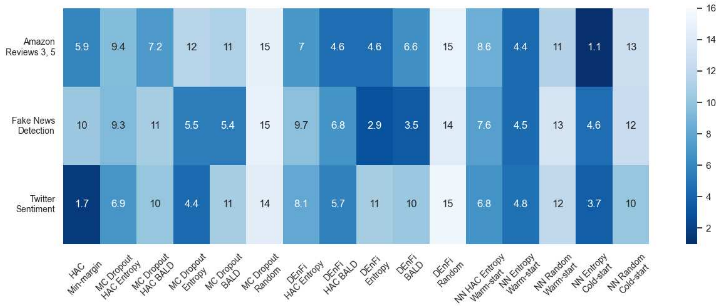


Figure 5: Aggregated mean rank for 14 tuples of learning algorithms and acquisition functions given Amazon Reviews 3,5, Fake News Detection, and Twitter Sentiment for 50 active learning iterations with batch size 20.

strategies that scale well with b. **Top** selects top b samples from sorted values of a_l. This approach may select samples close to each other, thus being redundant. **HAC** is a strategy based on the hierarchical clustering of a_l proposed in [8], and selecting top b samples from different clusters.

Tested algorithm variants:

From the range of all possibilities, we will study the combinations that exist in the literature: HAC min-margin using cold start [8], MC dropout with Entropy and BALD criteria using warm start [13], warm start ensemble learning with Entropy and BALD called DEnFi [27], and conventional single-network with prediction Entropy a with warm start. If HAC is not in the name, the Top strategy is used.

Since HAC strategy is an orthogonal factor to the remaining ones, we propose its combination with other approaches, giving rise to: HAC Entropy for the single neural network and both ensemble methods (MC dropout and DEnFi) and HAC BALD for the ensembles.

3 Experiment Setup

The methods were compared on different datasets and different batch sizes. The used datasets are positive/negative tweets from the Tweets [14], Fake News Detection [2], two pairs of categories from Amazon Reviews [**?**], and Gibberish [**?**] datasets. From all datasets, we select from 10000 text documents (5000 text documents per category, selecting only two categories for binary classification, e.g. 1 and 5 in Amazon reviews), which were the initial 10000 documents of the datasets given categories. The only exception is Fake News Detection where only 4000 documents are available (2000 text documents per category).

Experiment Parameters:

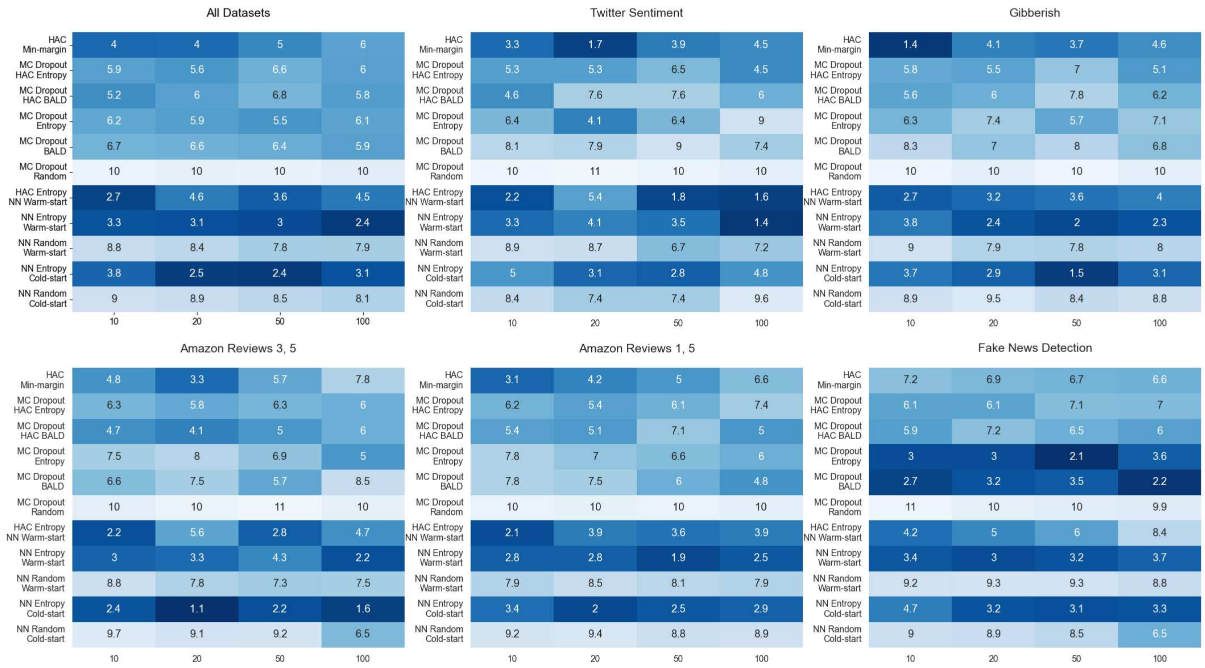


Figure 6: Aggregated rank for 7 active learning algorithms and two random strategies averaged over datasets as a function of different batch sizes.

Each active learning experiment was initialized by the training set $\mathbf{X}^{(0)}, \mathbf{Y}^{(0)}$ of 10 samples. The active learning strategy was set to sample b samples with a discrete set of variants, $b = 10, 20, 50, 100$. The active learning was run until 1000 samples were labeled, i.e. making a different number of steps for each batch size (10 iterations for $b = 100$, 20 for $b = 50$, etc.). The batch selection follows the ϵ -greedy approach [36], i.e. the samples selected by the acquisition function are accepted with probability $\epsilon = \frac{\exp(l-3)}{\exp(l-3)+1}$. A batch of random documents is selected for labeling if not accepted. The AUC is evaluated on the remaining part of the selected dataset (i.e. on the 9990 text documents in the first evaluation). The reported AUC values are averaged over 5 independent runs.

The initial number of epochs for the first iteration is 2500 for all algorithms. The same number is used for the cold start strategy in each iteration. The training of the warm start strategies is run for 150 epochs, with weights perturbation noise of variance $\sigma = 0.3$ for both MC dropout and DEnFi. Both DEnFi and MC Dropout generate 5 ensemble members. The key difference is in computational complexity, while DEnFi has to tune the parameters for each ensemble member, the MC dropout does it for only one network and generated ensemble members by 5 different realizations of the dropout mask.

Evaluation:

All algorithms were compared on the area under the curve (AUC) [12] on the test data (i.e. the documents not present in the training set). The algorithms were compared after 1000 acquired labels. The algorithms with smaller batch sizes thus benefited the from higher number of retrainings. To reduce the influence of stochastic initialization and training, the AUCs were run 5 times and averaged. Even then, the difference between the algorithms was sometimes marginal. To show the effect of various factors on the performance, we sorted the AUC and assigned a rank of each method accordingly. I.e. the best performing method has rank 1, second rank 2, etc. This approach allows the comparison of various methods across multiple datasets [9] using order statistics.

4 Results

Parameter uncertainty: The effect of parameter uncertainty (Bayesian approach) is the most costly to evaluate, due to the high computational demand of the ensemble approach (DEnFi). Therefore, we have evaluated all algorithm variants only for batch size $b = 20$. The results are displayed in Figure 5. The advantage of the Bayesian approach is evident only for the Fake News dataset. This behavior is a result of a good neural network parameters distribution estimate. However, in other datasets, DEnFi performed as good as a single neural network, and is not worth the computational cost. As a result, the algorithm was omitted from subsequent large-scale studies.

A summary of the performance of all tested methods for various batch sizes is displayed in Figure 4 via AUC

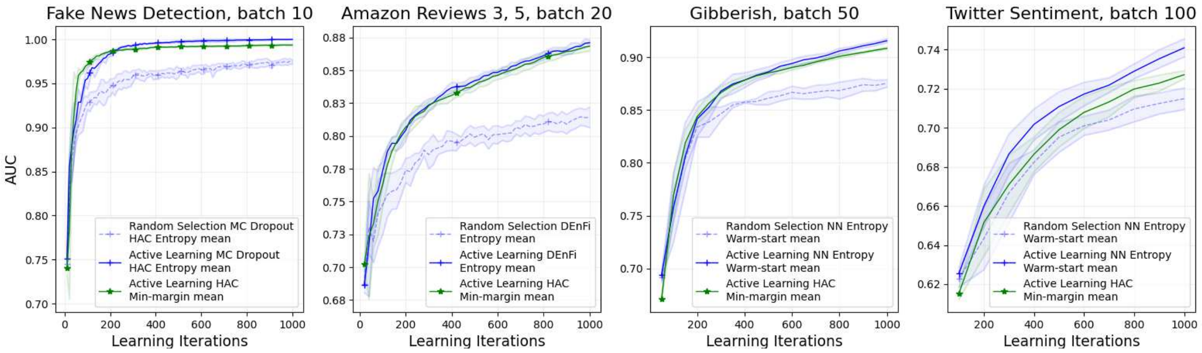


Figure 7: Evolution of the mean AUC with a growing number of requests for the best algorithms representative vs HAC Min-margin given the batch size and dataset. The uncertainty bounds are illustrated as one standard deviation from the mean value with respect to 5 runs. All algorithms were initially trained on 10 labeled text documents before sequential learning strategies were initialized

after 1000 samples for all methods, and via relative rank for all methods in Figure 6 averaged over ranks after each 100 label requests. Note that the datasets follow a similar pattern, with the exception of the Fake News data sets, where the parametric uncertainty (now represented only by the warm start MC dropout strategy) is beneficial, and HAC batch selection strategy has a negative effect (probably due to preference of large clusters).

Acquisition functions: Due to binary classification, the min-margin and entropy approaches coincide for a single network function. The difference between our generalization of Entropy and BALD methods for the ensemble techniques seems insignificant, Figure 6. HAC methods perform well for some cases, but a simple entropy approach shows more stable results for the majority of problems, batch sizes, and initializations.

Initialization of the training: The proposed modification on warm start strategies (HAC Entropy, NN Entropy, and NN Random warm start) are better or comparable in performance to the cold start (HAC Min-margin, NN Entropy, NN Random); this is achieved at a fraction of the training cost. This indicates that the additive noise is sufficient to avoid overfitting of the hot start [17].

Batch selection strategy: The HAC batch selection has a clear advantage for smaller batch sizes (10 and 20). This is consistent when comparing HAC and Top variants of all methods except Fake News Detection. Smaller batch sizes and the proposed generalization to warm start HAC outperforms the cold start approach in most of cases. The batch advantage diminishes for sizes of 50 and 100 where the top selection strategy achieves comparable (ensembles) or better (single NN) results. We project that the most informative samples in our datasets are clustered in small groups, hence the selection of a batch with a large enough size contains all important samples.

The contribution of this paper lies in a comparative study where we decomposed different algorithms into building blocks and generalized various approaches. For a better understanding of the methodology, we selected some approaches for demonstration. In Figure 7 a comparison of various active learning sequences is displayed. More specifically, the methods proposed by us to well studied HAC min-margin approach.

5 Conclusion

We have studied the influence of various factors (i. e. acquisition functions, batch sizes, neural networks initialization) of active learning algorithms and their performance on cover context integrity, fake news detection, and sentiment classification tasks. While complex algorithms such as deep ensembles (DEnFi and Dropout MC) sometimes achieve good performance (Fake News detection), the winner, on average, is the classical prediction entropy of a single neural network with a few proposed modifications like warm start. Although the performance of the warm start method can sometimes be the same assume the cold start, the undeniable benefit is a lower computational cost. The selection of the batch size for annotation is also important. The agglomerative clustering improves performance for smaller batch sizes and may show better results than a more general method like entropy sampling.

References

[1] Fake news dataset, 2018. `https://www.kaggle.com/c/fake-news`, accessed on 20 february 2021.

[2] Fake news detection dataset, 2018. `https://www.kaggle.com/jruvika/fake-news-detection`, accessed on 20 february 2021.

[3] Bang An, Wenjun Wu, and Huimin Han. Deep active learning for text classification. In *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, pages 1–6, 2018.

[4] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019.

[5] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

[6] Sreyasee Das Bhattacharjee, Ashit Talukder, and Bala Venkatram Balantrapu. Active learning based news veracity detection with feature weighting and deep-shallow fusion. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 556–565. IEEE, 2017.

[7] Sreyasee Das Bhattacharjee, William J Tolone, and Ved Suhas Paranjape. Identifying malicious social media contents using multi-view context-aware active learning. *Future Generation Computer Systems*, 100:365–379, 2019.

[8] Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 11933–11944. Curran Associates, Inc., 2021.

[9] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Liat Ein Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. Active learning for bert: an empirical study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, 2020.

[12] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[13] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

[14] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. Technical report, Stanford, 2009. `http://help.sentiment140.com/for-students/` accessed on 26 june 2020.

[15] Md Saqib Hasan, Rukshar Alam, and Muhammad Abdullah Adnan. Truth or lie: Pre-emptive detection of fake news in different languages through entropy-based active learning and multi-model neural ensemble. 2020.

[16] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[17] Peiyun Hu, Zachary C Lipton, Anima Anandkumar, and Deva Ramanan. Active learning with partial feedback. *arXiv preprint arXiv:1802.07427*, 2018.

[18] Pieter Floris Jacobs, Gideon Maillette de Buy Wenniger, Marco Wiering, and Lambert Schomaker. Active learning for reducing labeling effort in text classification tasks. In *Benelux Conference on Artificial Intelligence*, pages 3–29. Springer, 2021.

[19] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.

[20] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[22] David Lowell, Zachary C Lipton, and Byron C Wallace. Practical obstacles to deploying active learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 21–30, 2019.

[23] Katerina Margatina, Loic Barrault, and Nikolaos Aletras. Bayesian active learning with pretrained language models. *arXiv preprint arXiv:2104.08320*, 2021.

[24] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[25] Rishabh Misra. News category dataset, 06 2018. `https://www.kaggle.com/rmisra/news-category-dataset`, accessed on 26 june 2020.

[26] Sumanth Prabhu, Moosa Mohamed, and Hemant Misra. Multi-class text classification using bert-based active learning. *arXiv preprint arXiv:2104.14289*, 2021.

[27] Marko Sahan, Vaclav Smidl, and Radek Marik. Active learning for text classification and fake news detection. In *2021 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, pages 87–94. IEEE, 2021.

[28] Christopher Schröder, Andreas Niekler, and Martin Potthast. Uncertainty-based query strategies for active learning with transformers. *CoRR*, abs/2107.05687, 2021.

[29] Claude E Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[30] Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928*, 2017.

[31] Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. Byte pair encoding: A text compression scheme that accelerates pattern matching. Technical report, Citeseer, 1999.

[32] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13969–13980, 2019.

[33] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[34] Lukas Ulrych and Vaclav Smidl. Deep ensemble filter for active learning. Technical Report 2383, Institute of Information Theory and Automation, 2020.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[36] Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

[37] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.