Active Learning for Rumor Identification on Social Media

Parsa Farinneya[1], **Mohammad Mahdi Abdollah Pour**[1], **Sardar Hamidian**[2]
and **Mona Diab**[2, 3]

[1]Dep. of Computer Engineering, Amirkabir University of Technology
[2]Dep. of Computer Science, The George Washington University
[3]Facebook AI Research
{p_far, mabdollahpour}@aut.ac.ir, sardar@gwu.edu, mdiab@fb.com

Abstract

Social media has emerged as a key channel for seeking information. Online users spend several hours reading, posting, and searching for news on microblogging platforms daily. However, this could act as a double-edged sword especially when not all information online is reliable. Moreover, the inherently unmoderated nature of social media renders identifying unverified information ever more challenging. Most of the existing approaches for rumor tracking are not scalable because of their dependency on a significant amount of labeled data. In this work, we investigate this problem from different angles. We design an Active-Transfer Learning (ATL) strategy to identify rumors with a limited amount of annotated data. We go beyond that and investigate the impact of leveraging various machine learning approaches in addition to different contextual representations. We discuss the impact of multiple classifiers on a limited amount of annotated data followed by an interactive approach to gradually update the models by adding the least certain samples (LCS) from the pool of unlabeled data. Our proposed Active Learning (AL) strategy achieves faster convergence in terms of the F-score while requiring fewer annotated samples (42% of the whole dataset for the best model).

1 Introduction

Rumor detection in social networks is the task of identifying if a post's remark is unverifiable. This detection can help stop the spread of misinformation/dis-information that could potentially cause harm and distress. When a rumor about a subject emerges, there are thousands of posts shared about that subject. Ahsan (2019) show that having abundant in-domain labeled data can significantly impact the accuracy of the rumor detection model on Tweets by more than 30% improvement. This also points to the impact of out-of-domain/topic training on rumor detection performance. However in a real world scenarios for rumor detection, in domain human-annotated data is typically missing in early stages of rumor propagation, resulting in mediocre accuracy levels for such models. A viable solution for this problem would ideally be a framework that yields decent accuracy despite the absence of in-domain manually annotated training data. To this end, this paper proposes a semi-supervised framework based ATL for rumor detection in social media, specifically for Twitter data. There are three main variables for the proposed framework: the representation of the Tweets, the estimator, and the Active Learning strategy. Other experimental variables will be discussed in the following sections. As we evaluate all the different variables, we observe that Tweet-BERT, linear regression and least confidence strategy yield comparable results as non-Active Learning methods yet with a fraction of human-annotated data needed in non-Active Learning based methods. Further for robustness of our proposed models, we experiment with using an exploration method by choosing some random queries in each loop to prevent the model from overfitting. We also reach an approximation of the minimum labeled data needed for a decent classification in this task with the proposed method.

2 Related Work

Qazvinian et al. (2011) ran experiments to examine the effect of in-domain labeled data on rumor detection accuracy. They conducted learning curve experiments injecting their training models with labeled data, going from 400 to almost 2000 training examples. The experiments exhibit rapid performance improvement plateauing at an accuracy of 80%. Hamidian and Diab (2016) introduced the Tweet Latent Vector (TLV) feature, which is a 100-d vector that was created by a mixing Twitter features and network-specific features such as Hashtags, URL, Re-Tweets, and Content features such

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4556–4565
November 7–11, 2021. ©2021 Association for Computational Linguistics

as POS and content n-grams, as well as pragmatic features representing Named Entities, Sentiment. In 2019 ACL RumorEval shared task on rumor detection and verification, Derczynski et al. (2017) used a subset of the PHEME dataset in two subtask to identify the stance of comments as well as measure the veracity of the subset of rumor posts. The best models for this task utilized contextualized word embedding such as BERT. Additionally, the models used were mostly deep neural networks with the exception of the best performing model (Li et al., 2019), which was an ensemble of Support Vector Machine, Random Forrest and Logistic Regression. Bhattacharjee et al. (2017) proposed a simple, yet efficient, learning method for fake news detection in a weakly supervised scenario. The proposed method in this work improved generalization ability through interactive human participation by annotating a small amount of relevant samples that provide the most insightful information on the data. Their model was based on GloVe word embeddings and a CNN-based embedding model on the character level with fully connected layers for classification. They evaluated their models on the KDnugget's Fake News dataset[1], Liar Dataset, (Wang, 2017) and Harvard Dataverse Twitter Collection. Hasan et al. (2020) proposed an Active Learning framework for fake news detection based on entropy sampling. In this approach, by using just 4% to 28% of available training data, the model achieves a comparable performance to supervised learning with all available labeled training data. Inspired by this latter work, despite inherent differences in the task at hand (rumor detection vs. fake news detection), we believe that similar principles would hold for rumor detection. Accordingly, we propose a novel method for rumor detection that will reduce the need for human annotation in this task.

3 Problem Definition and Approach

3.1 Problem Definition

We cast the problem of rumor detection as a binary classification task. Tweets are classified as either rumors or non-rumor. We propose a human in the loop annotation strategy. When Tweets about a subject start spreading, and it is not clear whether it is a rumor, the proposed human-in-loop framework gradually trains a classification model specific for the emerged Tweet's subject. We propose a framework combining Active Learning with Transfer Learning. In each iteration of the proposed ATL pipeline, a batch of unlabeled Tweets that are the most informative for the model are passed to a human for annotation (similar to an Oracle in the Active Learning literature). This loop continues until the annotation budget is exhausted.

3.2 Active Learning

In this work, we leverage the most common Active Learning scenario that is the unlabeled pool scenario. This approach is also the most similar to real-life problems. In this scenario, there is a large pool of unlabeled data. The model is at first trained on a small subset of pre-annotated data. Then the framework queries for a batch of unlabeled data to be labeled by a human (oracle) and added to the train set on each iteration. Since annotation may be expensive or time-consuming, it is preferable to run this process as few times as possible. The sample queries are chosen among unused unlabelled data based on their score, and the scoring function is called the strategy in the Active Learning literature. This step is repeated until the annotation budget is exhausted. The algorithm is described in Algorithm1.

Various strategies are proposed in the literature for data selection in an Active Learning pipeline. Selection based on prediction uncertainty is the most popular approach, which is also applied in this work.

Least Confidence (LC) Least Confidence (LC) is a strategy based on prediction uncertainty. LC tries to find data samples that the model is not certain about, as a proxy for the model having trouble classifying that data. Certainty is measured as confidence in most likely label as defined in the equation 1 by $max(y)$. y is probabilities predicted by the model given x as input and $score(x)$ is the uncertainty measure.

$$score(x) = 1 - max(y) \tag{1}$$

Query by Committee (QBC) In Query by Committee (QBC) strategy, instead of measuring the uncertainty of a single model, we train an ensemble of models. For a given sample, disagreement between models is taken as a measure of uncertainty. There are also two special cases of QBC: bagging (BAG) and boosting (BOOST). In BOOST, we bootstrap random samples with replacement from the available initial data for the committee members. In

[1] https://github.com/lutzhamel/fake-news

```
Algorithm 1: Pool-based Active Learning
  Input: $D_i$, $D_p$, $D_{te}$, batch size, strategy,
         estimator, annotation budget
  Output: Model, metrics
  $D_i$ Initial data;
  $D_p$ Pool data;
  $D_{te}$ Test data;
  Instantiate $D_{tr}$ as empty, Train data;
  Instantiate model;
  Add $D_i$ to $D_{tr}$;
  model = estimator.train($D_{tr}$);
  while annotation budget is not over do
      $D_q$ = Query($D_p$, batch size, strategy,
        model);
      Remove $D_q$ from $D_p$;
      Annotate $D_q$;
      Add $D_q$ to $D_{tr}$;
      train model on $D_{tr}$ from scratch;
      Compute and save metrics;
  end
```

BAG, we perform bootstrapping for both initial and train data.

Ranked Batch (Batch-LC) We also use the Ranked Batch strategy (Batch-LC) as proposed in Cardoso et al. (2017) which uses a scoring function as in Equation 2 to find a ranked list of query data.

$$score = \alpha(1-\Phi(x, X_{labeled}))+(1-\alpha)U(x) \quad (2)$$

In Equation 2, $X_{labeled}$ is the labeled dataset, $U(x)$ is the uncertainty of predictions for x, and Φ is a similarity function, for instance, cosine similarity. This latter function measures how well the feature space is explored near x. α is also computed by Equation 3.

$$\alpha = \frac{|X_{unlabeled}|}{|X_{labeled}| + |X_{unlabeled}|} \quad (3)$$

After score computation for each sample, the highest scoring sample is removed, and scores are recalculated until the desired number of examples are available to send for the query.

Epsilon-Greedy (EG) In order to find a balance between exploration and exploitation, we use a method inspired by ϵ-greedy (EG) strategy in Reinforcement Learning. We implement this approach in two ways: inter-batch and intra-batch. Inter-batch EG (EG-inter) selects query data at each iteration randomly with probability ϵ otherwise chooses the data based on LC $(1-\epsilon)$. Intra-batch (EG-Intra) dedicates $\epsilon\%$ of query data to RND and $1-\epsilon\%$ of that to LC. We use $\epsilon = 0.2$ (20% in EG-intra) in our experiments.[2]

3.3 Cross Topic Transfer

Data from other domains can be beneficial to improve performance on the target domain. Therefore, we design another type of experiment in which Tweets from other topics are considered in the initial feed to the model (zero shot). The model queries the pooled data at each iteration from the target topic. This is the setting that usually appears in real-world problems. There are datasets from previous topics that can not generalize well to the target topic, However, by choosing a minimum number of data through Active Learning, the model can adapt to the target domain.

4 Experimental Setup

We compare each experiment with Least Confidence (LC) and random strategy (RND) leveraging different representations, and learning algorithms. In random strategy, data samples are chosen uniformly random at each iteration. To mitigate the effect of randomness in both strategies, training algorithms and data splits, for each experiment, we randomly split the dataset into two sections, first one for initial and pool data and the second one as test data. We do this 5 times, and average the results (in a cross validation type evaluation strategy). For each of these 5 runs, the splits are the same among different experiments.

At each iteration, Multi-Layer Perceptron (MLP) is retrained on the batch of data annotated in the current iteration since training from scratch would be computationally expensive. However, other models are trained from scratch on data that was obtained in the current and previous iterations. In all experiments, we train the models on 20 randomly chosen samples as the initial dataset and query for 50 samples at each iteration, i.e. batch size = 50.

After examining all the models and representation settings with LC and RND strategy, the best setting is utilized for further experiments leveraging other sampling strategies such as Query batched committee (QBC), Epsilon-Greedy (EG), and Ranked Batch (Batch-LC).

[2] Values have been determined empirically on a tuning set.

4.1 Representation

We use SOTA representations for this task, such as BERT (Devlin et al., 2019) and TweetBERT (Qudar and Mago, 2020). TweetBERT, a domain-specific BERT based language model trained specifically on social media data. TweetBERT was trained on about 680 million Tweets . We also use earlier representations such as GloVe (Pennington et al., 2014). For each sentence, we average the GloVe vector representation of all tokens in the sentence and use it as the input to the models. We use Twitter GloVe, which is consistent with the domain of our work. Twitter GloVe was trained over by 2B Tweets, 27B tokens, and 1.2M vocab. A dimensionality of 200 was determined empirically to yield best results.[3] The representations are frozen during training.

4.2 Model

We also examine different models that have been mainly used for short text classification tasks, namely, MLP (Hinton, 1990), Support Vector Machines (SVM) (Platt et al., 1999), Random Forests (RF) (Breiman, 2001), Logistic Regression (LR) (Cramer, 2002), Ada boosted decision trees (Ada) (Freund and Schapire, 1997), K-Nearest Neighbors (KNN) (Fix, 1985), Gaussian Process Classifier (GP) (Rasmussen, 2003), Linear Discriminant Analysis (LDA) (Cohen et al., 2003), and Quadratic Discriminant Analysis (QDA) (Tharwat, 2016).

We used Radial Basis Function (RBF) kernel for SVM with inverse regularization term $C = 1$. For Random Forest, we used 100 estimators and a max depth of 1000 with the Gini criterion. For Logistic Regression, we used l_2 penalty and LBFGS (Liu and Nocedal, 1989) solver with a maximum of 100 iterations. For Ada boosted decision tree, we used an ensemble of 50 trees with SAMME.R real boosting (Freund and Schapire, 1997). The MLP had a hidden layer of size 128 and a drop-out layer after the hidden layer with $p = 0.3$ and it was trained by adam optimizer (Kingma and Ba, 2015). We use two KNN models: one with 5 neighbors (KNN5) and the other with 3 neighbors (KNN3). GP is used with an RBF kernel and optimized with the L-BFGS-B (Byrd et al., 1995) algorithm. An SVD solver was used for both LDA and QDA.

Machine Learning Tools There are some tools and libraries used to build the experimental pipeline. MLP was implemented in Tensorflow[4] and for other models (RF, SVM, LR, Ada, KNN, GP, LDA and QDA) we used Scikit-Learn (Pedregosa et al., 2011) package. Active Learning workflows were developed using modAL (Danka and Horvath, 2018) framework.

4.3 Data

The PHEME dataset is curated from highly retweeted Tweets associated with newsworthy events (Zubiaga et al., 2016). It includes five cases of breaking news: *Ferguson unrest*, *Ottawa shooting*, *Sydney Siege*, *Charlie Hebdo shooting*, and *Germanwings plane crash*. It also includes four specific rumors: *Prince to play in Toronto, Gurlitt collection, Putin missing, and Michael Essien contracted Ebola*. This dataset consists of 6425 Tweets comprising 2402 rumors, and 4023 non-rumors. In this study, we work on *Charlie Hebdo*, *Ferguson*, and *Sydney Siege* since they have the highest number of annotated Tweets in the dataset (more than 1000 Tweets each). In topics with a small number of Tweets, it is not possible to have an unbiased test set and examine the effect of AL on choosing a minimum number of data. For example for a topic with only 100 labeled tweets in dataset, we can not have a reliable test set (at least 1000 samples) and a big pool dataset (100 samples are consumed by AL in 2 iterations)

Preprocessing The texts of Tweets were processed by changing "'t" to "not", for privacy and generalization changing usernames to "Username", removing punctuation except question marks, removing special characters, removing trailing white space, and changing URLs to "Link". Table 1 shows some samples of this data set.

4.4 Metrics

We evaluate the performance of models by F1 score instead of accuracy since the test data does not come from a distribution with balanced labels. Moreover, we examine the effectiveness of Active Learning through some additional metrics. For each setting, we compute the F1 score variation in Active Learning loops. Namely, we compute F1 score on points which account for using 0%, 25%, 50%, 75%, and 100% of the pool data.

[3] To select the right GloVe dimension, we experimented with all 4 sizes of GloVe 25, 50, 100 and 200, the latter yielded the best results across the board.

[4] https://www.tensorflow.org/

Charlie Hebdo	
Rumors	Non-rumors
#CharlieHebdo witness - Gunmen told me to tell the media they were Al-Qaeda in Yemen	Just arrived at scene of massacre #Paris #charliehebdo
According to #CharlieHebdo\u2019s lawyer four well-known French cartoonists were killed by the masked gunmen: Cabu, Wolinski, Charb et Tignous.	Anybody who wants to talk about what Charlie Hebdo might have done to \"provoke\" this should probably shut up, forever

Table 1: Tweet samples of PHEME dataset

In order to determine the minimum amount of data needed for each experiment to achieve a promising result, we consider a minimum number of data samples needed to achieve at least $f_{max} - 1\%$ where f_{max} is the maximum F1 score reached in that experiment.

5 Experimental Results

5.1 Baselines

We compare the models against two baselines: RANDOM and Majority based on training data observations. RANDOM is simply random prediction. Majority is simply the majority of labels observed in the training data at each iteration are predicted for all samples in the test set.

5.2 Estimator selection

The proposed method of Active Learning has a base estimator that estimates pool data and predicts the test set. Topics that weren't used in Active Learning loops due to having few Tweets, Ottawa shooting, Germanwings, were used for hyper-parameter tuning in a greedy search base method.

5.3 Results

Tables 2 shows F1 score at points of using 0%, 25%, 50%, 75%, and 100% of pool data for each experiment. In each column scores go from red to white and green as models consume more data showing how rapidly the model improves.

5.3.1 Model and Representation Comparison

By examining the results, Logistic Regression yields the best performance among our models, and TweetBERT is the best representation. This is expected since TweetBERT is pretrained on the tweets genre. Interestingly, GloVe representations outperform BERT respresentations, despite the fact that BERT is known for its more sophisticated architecture yielding contextualized embeddings. However, our version of GloVe embeddings is trained on Twitter data. This observation suggests that the genre of the training data has a larger impact on performance than the representation model complexity. TweetBERT+LR with LC strategy achieves the best scores. Experiments with LC strategy perform better than RND. TweetBERT+LR with LC strategy also achieves the best performance with only 25% of data.

5.3.2 Strategy Comparison

We examine the performance gain of LC in more detail by comparing the difference of F1 score for RND strategy and LC strategy of a fixed setting at points of using 0%, 25%, 50%, 75%, and 100% of pool data for each experiment. Table 2 illustrates some of these observations. For instance, rows 0% and 100% in the table shows where the model has access to same portion of data, whether using LC or RND. Subtracting values in RND column from LC column for LR with TweetBERT yields 0, 2.07, 2.2, 1.33 and 0.03 for each row, respectively. Similarily for rows 25%, 50%, and 75%, we observe the effect of Active Learning such that the differences for most of model-representation pairs would be positive, indicating an improvement over the RND strategy. RF, SVM, and GP get the benefit the most from LC strategy.

Table 4 compares the performance of uncertainty strategies using best representation-model pairs. Except for QBC, other strategies are very close. Based on our results, Ranked Batch (Batch-LC), boosting (BOOST), and bagging (BAG) yield the best performance, respectively. QBC fails to make a diverse ensemble but when used with BOOST and BAG there are more diverse voters. In each row scores go from red to white and green as models consume more data showing improvement of models.

Figure 1, 2 and 3 show F1 score for Tweet-

	Model	Ada		GP		KNN3		KNN5		LDA	
	Stra.	LC	RND	LC	RND	LC	RND	LC	RND	LC	RND
TweetBERT	0%	58.6	58.83	60.6	60.6	64.3	64.3	62.73	62.73	59.83	59.83
	25%	70.37	70.43	72.57	63.5	70.2	69	71.57	70.27	68.8	67.83
	50%	72.2	72.6	76.87	68.2	73.6	73.33	75.03	74.3	68.4	67
	75%	74.1	73.6	75.37	70.67	75.33	74.97	76.1	74.93	64.53	62.6
	100%	74.17	74.07	72.3	72.23	75.87	75.83	76.2	76	63.93	63.77
GloVe	0%	60.37	59.53	58.33	58.33	64.7	64.7	64.23	64.23	63.6	63.6
	25%	70.2	70.53	73.47	66.33	72.3	71.27	73.03	70.9	64.7	61.23
	50%	73.4	72.9	77.97	71.83	75.5	75.33	75.87	75.03	66.1	67.23
	75%	74.7	73.77	76.2	74.37	76.33	76	76.67	76.33	73.37	73.1
	100%	73.9	74.07	75.37	75.4	76.83	76.97	77.37	77.53	76.8	76.67
BERT	0%	56.93	57.53	58.67	58.67	55.73	55.73	52.27	52.27	55.67	55.67
	25%	67.6	65.97	64.63	66.97	66.47	66.77	65.8	66.73	70.57	71.07
	50%	70.43	68.93	68.1	68.23	69.87	69.67	70.27	69.87	70.67	69.43
	75%	71.23	70.97	71.4	70.03	71.03	70.57	71.63	71.53	69.53	66.5
	100%	70.5	71.67	71.33	71.27	71.57	71.33	71.83	71.77	63.5	63.33
	Model	LR		MLP		QDA		RF		SVM	
	Stra.	LC	RND	LC	RND	LC	RND	LC	RND	LC	RND
TweetBERT	0%	60.77	60.77	41.77	41.77	50.17	50.17	56.3	56.3	43.1	43.1
	25%	76.6	74.53	56.47	55.63	43.83	40.2	74.33	67.73	43.53	41.73
	50%	78.6	76.4	75.63	69.43	28.13	32.73	75.13	70.27	54.53	55.17
	75%	**78.93**	77.6	69.57	71.87	34.93	35.83	73.5	71.37	63.7	61.03
	100%	78.5	**78.47**	71.67	74.17	37.43	38.13	72.03	72.4	65.93	66.03
GloVe	0%	59.4	59.4	45.37	42.97	51.67	51.67	58.87	58.87	50.17	50.17
	25%	75.9	73.13	56.93	57.77	47.5	44.17	74.4	67.67	69.17	61.37
	50%	78.47	76.1	74	70.37	38.9	46.03	76.23	71.13	75.83	71.2
	75%	78.2	77.53	75.07	74.13	42.27	42.33	74.67	72.7	76.3	75.43
	100%	78.27	78.27	76.83	76.7	47.97	47.7	74.3	74.03	77.47	77.3
BERT	0%	58.27	58.27	46.63	43.83	49.7	49.7	53.47	53.47	43.1	43.1
	25%	73.33	72.5	64.73	65.73	51.9	50.57	70	58.3	65.07	45.53
	50%	76.87	74.57	72.03	69.97	50.6	51.53	68.87	63.27	62.7	54.87
	75%	76.77	75.53	73.5	72.93	52.37	52.37	66.9	64.8	63.2	60.67
	100%	76.83	76.77	73.4	73.57	49.2	50.03	66.6	66.57	64.13	64.1

Table 2: F1 score at points of using 0%, 25%, 50%, 75%, and 100% of pool data for each experiment with all representations and model representations. The scores are averaged over the three chosen topics in the PHEME dataset. Baseline RANDOM prediction baseline achieves 26.98% F1 score, and Baseline Majority prediction baseline achieves 40.90±3.3%. Intensity of color green shows high F1 scores, red show low F1 scores and white for inbetween F1 scores.

Approach	0%	25%	50%	75%	100%
Few Shot	60.767	76.6	78.6	78.933	78.5
Zero Shot	50.1	57.5	65.867	69.6	71.033

Table 3: F1 score for TweetBERT+LR with LC strategy using 0%, 25%, 50%, 75% and 100% of pool dataset. The scores are averaged over three chosen topics in PHEME dataset. In the Zero Shot setting, the initial dataset includes other topics, as opposed to the Few Shot setting, in which, initial training data consists of 20 samples of the target topic. Intensity of color green shows high F1 scores, red show low F1 scores and white for inbetween F1 scores

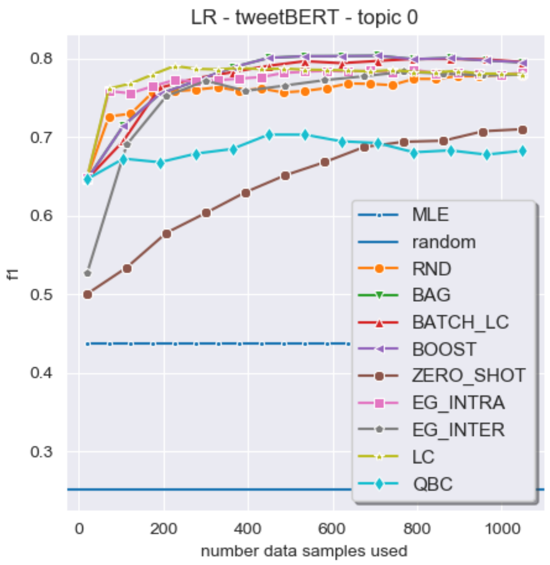
Representation	Model	Strategy	0%	25%	50%	75%	100%
TweetBERT	LR	QBC	64.7	66.8	70.3	69.3	68.3
		BAG	60.7	75.4	79.067	79.533	79
		Batch_LC	64.767	76.4	78.8	79.267	79.167
		BOOST	60.7	75.4	79.3	79.767	79.033
		EG_intra	64.9	77.533	78.167	78.3	78.133
		EG_inter	52.7	75.2	77.133	78.267	78
		LC	60.767	76.6	78.6	78.933	78.5
GloVe	LR	QBC	65.2	65.3	68.2	67.3	67.6
		BAG	59.933	75.1	78.733	78.6	78.767
		Batch_LC	64.133	75.6	78.967	78.867	78.8
		BOOST	59.933	75.1	78.7	78.6	78.767
		LC	59.4	75.9	78.467	78.2	78.267

Table 4: F1 score for advanced strategies of a fixed setting at points of using 0%, 25%, 50%, 75% and 100% of pool data for best representation-estimator pairs. The scores are average over three chosen topics in PHEME dataset. Intensity of color green shows high F1 scores, red show low F1 scores and white for inbetween F1 scores.

BERT+LR with different uncertainty strategies. The diagrams indicate that most models plateau with 100-200 data samples and are able to achieve decent performance with a small amount of data. Most models have a large gain with 100-200 (well-chosen with AL) data samples and there is a small gain after having more than 200 samples. Our best model (TweetBBERT+LR with BATCH-LC) achieves at least $f_{max} - 1\%$ with 250, 300 and 250 data samples from pool data for each topic. (f_{max} being maximum of F1 score reached in that experiment) On average, it achieves at least $f_{max} - 1\%$ with 42% of pool data (There are 1039, 571, 610 samples in pool dataset of each topic).

5.3.3 Cross-Topic Evaluation

Table 3 compares best performing model-representation pair with LC strategy starting from two different initial training datasets. The initial training dataset in zero-shot approach is all data for all topics except the target topic. The initial training dataset of few-shot approach contains a minimal number of in topic in domain samples. We experiment with 20 samples of the target topic. We observe that only a few topic-related samples perform much better than a large dataset of samples, namely, the few shot setting outperforms the zero shot setting as observed in the 0% of pool data column in Table 3. Data from other domains/topics causes a high variance, which takes many related samples for the model to converge onto reasonable performance. The results demonstrated that Tweets from other rumor topics can add some bias to the model and make the model degrade in performance.

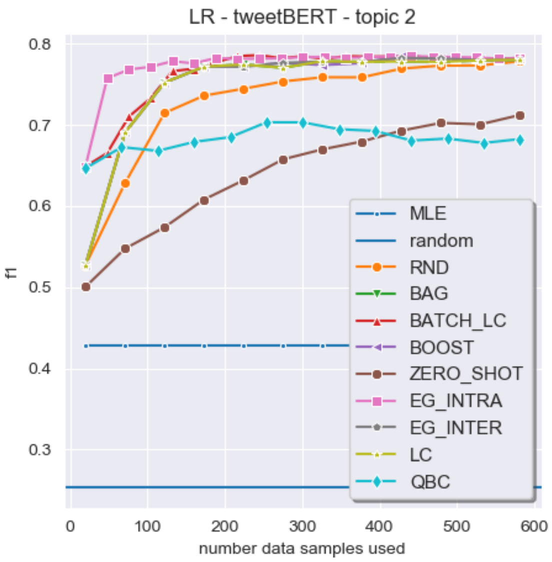


cc

Figure 1: Performance of different strategies with TweetBERT+LR on the *Charlie Hebdo* topic. The Verticals axis shows F1 score and the horizontal axis shows number of data samples used for training during Active Learning.

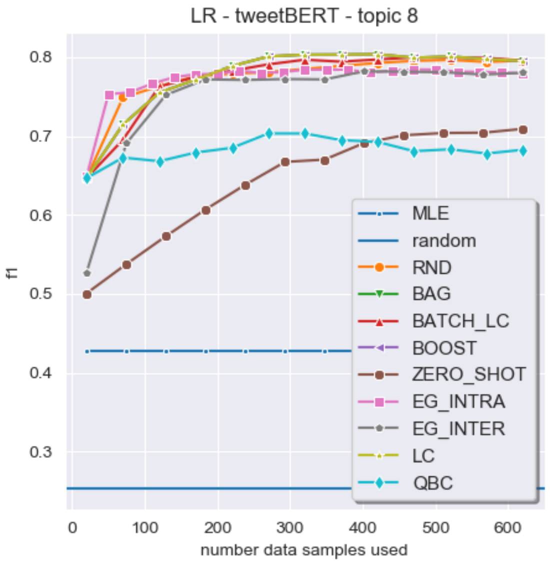
6 Error Analysis

The confusion matrix for TweetBERT+LR with EG-intra strategy on the *Sydney Siege* topic for different steps is shown in Table 5. We see that the performance gain is majorly the result of decreasing false negatives. Confusion matrix for other topics also showed a similar behaviour. The model ability to detect rumors improves with the amount of data compared to ability to detect non-rumors. Since, model is able to encode better boundaries for rumors, while non-rumors might be diverse.



cc

Figure 2: Performance of different strategies with TweetBERT+LR on the *Sydney Siege* topic. The Verticals axis shows F1 score and the horizontal axis shows number of data samples used for training during Active Learning.



cc

Figure 3: Performance of different strategies with TweetBERT+LR on the *Ferguson* topic. The Verticals axis shows F1 score and the horizontal axis shows number of data samples used for training during Active Learning.

	TN	FP	FN	TP
0%	44.42	12.14	19.40	24.02
25%	45.32	11.24	10.88	32.54
50%	46.58	9.98	10.38	33.04
75%	47.25	9.31	10.21	33.21
100%	47.45	9.11	10.88	32.54

Table 5: Confusion matrix on *Sydney siege* topic, averaged over 5 runs. Numbers in the columns are percentages of True Negatives (NT), False Positives (FP), False Negatives (FN) and True Positives (TP), respectively. These numbers of average of number in 5 runs. The First column shows percentage of pool data consumed by the model.

7 Conclusion & Future Directions

We proposed an active-transfer learning framework for the rumor detection task. In our proposed framework, we examined different word representations, estimators and Active Learning strategies. More than 300 experimental setups were run and each setup was fine tuned to yield the best results. Our experiments indicate multiple new findings: 1. The approximate minimum number of labeled in-domain data needed for a decent rumor detection model with our proposed method is around 200; 2. In-genre pretrained (contextualized) LMs have the biggest impact on model performance; 3. We investigate and empirically show how epsilon-greedy inspired methods that joins randomness and uncertainty in query selection could prevent the model from over-fitting; and, 4. We also showed that naive use of Tweets relating to other topics can degrade the performance (the zero shot setting). Although the method proposed in this paper did not show improvement in using data from other topics, information from different topics can be exploited by incorporating other techniques such as weighting the data samples or meta-learning few-shot domain adaptation. Diverse initial datasets may yield an initial model with better uncertainty scores and earlier convergence. The next step for this method would be incorporating metadata such as reply stances, user information, network propagation information, etc. Finally, another method that could improve our proposed model would be using an ensemble of different representations and different models to generalize better.

8 Ethical Considerations

8.1 NLP Application

Misuse Potential and Failure Mode When used as intended, applying the strategy described in this paper can help to use the minimum amount of labeled data to identify new emerging rumors online. However, the annotation volume might be inconsistent in some rumors with high variants. This may lead to Failure and high bias. Further research is needed to address the rumor identification issues for emerging rumors, as this issue is present among all current methodologies.

Environmental Cost The experiments described in the paper use a single CPU for all the machine learning models except MLP, which used GPUs. The experiments may take several hours. Several dozen experiments were run due to parameter search for all the models, and future work should experiment with distilled models for more lightweight training. We note that while our work required extensive experiments to draw sound conclusions, future work will be able to draw on these insights and need not run as many large-scale comparisons. Models in production may be trained once for use using the most promising settings.

References

Mohammad Ahsan. 2019. Detection of context-varying rumors on twitter through deep learning. 128:45–58.

Sreyasee Das Bhattacharjee, Ashit Talukder, and Bala Venkatram Balantrapu. 2017. Active learning based news veracity detection with feature weighting and deep-shallow fusion. In *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017*, pages 556–565. IEEE Computer Society.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208.

Thiago NC Cardoso, Rodrigo M Silva, Sérgio Canuto, Mirella M Moro, and Marcos A Gonçalves. 2017. Ranked batch-mode active learning. *Information Sciences*, 379:313–337.

Jacob Cohen, Patricia Cohen, Stephen G West, and Leona S Aiken. 2003. Applied multiple regression. *Correlation Analysis for the Behavioral Sciences*, 3.

Jan Salomon Cramer. 2002. The origins of logistic regression.

Tivadar Danka and Peter Horvath. 2018. modAL: A modular active learning framework for Python. Available on arXiv at `https://arxiv.org/abs/1805.00979`.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Evelyn Fix. 1985. *Discriminatory analysis: nonparametric discrimination, consistency properties*, volume 1. USAF school of Aviation Medicine.

Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Sardar Hamidian and Mona Diab. 2016. Rumor identification and belief investigation on Twitter. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 3–8, San Diego, California. Association for Computational Linguistics.

Md Saqib Hasan, Rukshar Alam, and Muhammad Abdullah Adnan. 2020. Truth or lie: Pre-emptive detection of fake news in different languages through entropy-based active learning and multi-model neural ensemble. pages 55–59.

Geoffrey E Hinton. 1990. Connectionist learning procedures. artificial intelligence, 40 1-3: 185 234, 1989. reprinted in j. carbonell, editor,". *Machine Learning: Paradigms and Methods", MIT Press.*

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Quanzhi Li, Qiong Zhang, and Luo Si. 2019. eventAI at SemEval-2019 task 7: Rumor detection on social media by exploiting content, user credibility and propagation information. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 855–859, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Dong C Liu and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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.

Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Mohiuddin Md Abdul Qudar and Vijay Mago. 2020. Tweetbert: A pretrained language representation model for twitter text analysis. *arXiv preprint arXiv:2010.11091*.

Carl Edward Rasmussen. 2003. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer.

Alaa Tharwat. 2016. Linear vs. quadratic discriminant analysis classifier: a tutorial. *International Journal of Applied Pattern Recognition*, 3(2):145–180.

William Yang Wang. 2017. “liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PLoS ONE*, 11(3).