Detecting Social Bots on Twitter: An Anytime Active Learning Approach [*]

Maria E. Ramirez-Loaiza
mramire8@hawk.iit.edu
Illinois Institute of Technology
Chicago, IL 60616

Abstract

Social networks allow the general publish to generate, share and spread information at a fast pace. However, even though spreading information fast can be beneficial, it can also increases the risk of misinformation due to lack of validation. On Twitter, automated accounts can easily increase the traffic of unvalidated information. There has been research efforts in machine learning to detect such automated content generator, to help identify reliable sources. We propose to build a machine learning model using active learning strategies to reduce the effort of annotation. To the best of our knowledge this is a novel method to address the classification of human and automated Twitter accounts.

1 Introduction

`Outline: Use in social networks`

The advent of social networks provided the general public with outlets to communicate, express ideas and sentiment, and spread information at fast rates and vast volumes. The emergent property of instance communication has been use to keep contact when other ways are not possible, for example, during natural disasters families can communicate. However, there is downside of the same property. Much of the information spread through the media is not verified and sources are unknown, causing the spread of misinformation. For example, a user can broadcast a message and quickly spread around the world without being verified. In a recent report [1], Twitter estimated that 8% (about 23 million accounts) of their active monthly users have no discernible human control, i.e., there is not way to determine if the there is a human producing the information. This undetermined accounts are usually referred as *social bots* or simply *bots*

There have been efforts to develop supervised machine learning methods to determine if an account is likely to be from a human or not. However, this methods typically require a lot labeled data, and the labeling effort can be costly. *Active learning* is an alternative to build annotated data and reduce the human effort on annotation. Typically, an active learner iteratively selects unlabeled examples and queries the labels to a human annotator or *oracle*. In this paper, we propose the use of active learning methods to speed up annotation of Tweeter data to detect social bots. We begin with the intuition that during annotation a human annotator will look for a key tweet that provides evidence of bot generated content.

The main idea is to select a user's content and pick a tweet to query the oracle for the label. The question is whether we can find the best tweet for annotation that will allow the oracle to provide a correct label. The saving in annotation comes from the oracle analyzing only one carefully select tweet instead of a full timeline. The drawback is that the oracle can incorrectly label a tweet and introduce errors in the training data. Therefore, the active learner has to

[*] This report was developed as part of a course project for CS579-F14 and is part of the AAL research project at the Machine Learning Lab at IIT.

[1] From the Twitter 2Q 2014 Earnings Report available on https://investor.twitterinc.com

consider what tweet will provide the best change of correct annotation.

In this work, we evaluate several approaches to select tweets. We performed experiments on collected data based on an existing list of known human and bot accounts (Lee et al., 2011). Our research questions are as follows:

RQ1. How does the data representation affect the classification performance? We found that for this classification task the tested representation options do not affect the classification accuracy significantly.

RQ2. How does the model selection affect active learning performance? We found that the best models are multinomial naive bayes and logistic regression with L2 regularization. However, during the active learning loop, the best results are obtained by logistic regression.

RQ3. How does selecting the best tweet affect the learning efficiency? We found that selecting the best tweet is not able to outperform the first and random tweet baselines. We provide insights regarding the possible cause of the poor performance.

2 Methodology

In this section, we first review standard active learning and then formulate our proposed method.

2.1 Problem Formulation

Let $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ be a labeled dataset where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector that represents a Twitter user timeline and $y_i \in \{y^0 = \text{human}, y^1 = \text{bot}\}$ is its class label. Let $\mathcal{U} = \{\mathbf{x}_i\}_{i=l+1}^{m}$ be a set of unlabeled examples. Let $P_{\mathcal{L}}(y|\mathbf{x})$ be the conditional probability of y given $\mathbf{x}$ according to a classifier trained on $\mathcal{L}$.

Typical pool-based active learning selects instances $\mathcal{U}^* \subseteq \mathcal{U}$ to be labeled by a human annotator (*oracle*) and appended to $\mathcal{L}$. Assuming a prespecified annotation budget B and an annotation cost function $C(\mathbf{x})$, the goal of the active learning algorithm (*student*) is to select $\mathcal{U}^*$ to minimize the classifier's generalization error subject to the budget constraints:

$$\mathcal{U}^* \leftarrow \operatorname*{argmin}_{\mathcal{U}_i \subseteq \mathcal{U}} Err(P_{\mathcal{L} \,\cup\, \mathcal{U}_i}(y|\mathbf{x})) \quad \text{s.t.} \sum_{\mathbf{x}_j \in \mathcal{U}_i} C(\mathbf{x}_j) \leq B \qquad (1)$$

Equation 1 is typically optimized by greedy algorithms, selecting one or more examples at a time according to some heuristic criterion that estimates the utility of each labeled example. A common approach is to request a label for the unlabeled instance that maximizes benefit-cost ratio: $\mathbf{x}_i^* \leftarrow \arg\max_{\mathbf{x}_i \in \mathcal{U}} \frac{U(\mathbf{x}_i)}{C(\mathbf{x}_i)}$.

Various definitions of utility $U(\cdot)$ are used in the literature, such as expected error reduction Roy and McCallum (2001) and classifier uncertainty Lewis and Gale (1994). In this paper, we use uncertainty sampling for our formulation. More formally, uncertainty sampling queries the instances whose predicted posterior probability is the least confident, redefining Equation 1:

$$\mathbf{x}^* \leftarrow \arg\max_{\mathbf{x}_i \in \mathcal{U}} \left(1 - \max_{y \in Y} P_{\mathcal{L}}(\hat{y}|\mathbf{x}_i)\right) \qquad (2)$$

Equation 2 uses conditional error as a measure of confidence.

We propose the use of an anytime active learning technique to save annotation cost by controlling what the oracle sees (Ramirez-Loaiza et al., 2014). The idea is that the student selects the most representative tweet of a user timeline and queries the oracle for the label. We define an alternative formulation of the active learning problem in which the student has the added capability of presenting one tweet the human oracle to request a label.

Let s_i^k be the k-th tweet in user $\mathbf{x}_i$ timeline. Each tweet is scored by a function $TS(\cdot)$ which determines how important is a tweet for labeling. We build upon Equation 2 and incorporate the tweet selection into the student's objective:

$$\arg\max_{\mathbf{x}_i \in \mathcal{U}} 1 - \max_{y \in Y} P_{\mathcal{L}}(\hat{y}|\mathbf{x}_i) \times max_{s_i^k \in \mathbf{x}_i} TS(s_i^k) \qquad (3)$$

In our experiments, we define $TS(\cdot)$ as the tweet most likely to be labeled. Intuitively, we want to select a tweet that allows the oracle to determine if the author of the tweet is human or not. Also, we want that tweet to be likely to be labeled since we optimize the budget simultaneously. Formally, we define the function as:

$$TS(s_i^k) = \max_y T(y_i|s_i^k) \tag{4}$$

where $T(\cdot)$ is a tweet probabilistic classifier that determines how likely is an individual tweet to be bot or human generated. In practice, we use tweets $s_i^k \in \mathbf{x}_i \in \mathcal{L}$ to train the tweet classifier.

3 Experimental Evaluation

3.1 Data Collection

Our experiments uses data collected from Twitter based on the **Social Honeypots** dataset Lee et al. (2011). We use a subset of a collection of known legitimate and bot account user names, to build our own dataset. We collected the most recent timeline of each user up to 200 tweets. We collected 883 legitimate user and 898 bot accounts.

The data was further filtered by removing accounts whose last activity was older than 2014, eliminating 45 and 173 accounts from legitimate and bots respectively. For our active learning experiments, we report average of five trials on a train-test split.

3.2 Data Preprocessing

Each tweet was tokenized by words, ignoring punctuation, collapsing URLs and mentions. Every token was converted to lower case and stemmed using a Porter stemmer. The resulting tokens were used to form a TF-IDF feature vector using unigrams. To reduce the size of the dictionary terms that appear less than five times are removed. The final dictionary size is of 13399 features. Table 1 characteristics of the representation.

Table 1: Collected data from Twitter per type of user

Class	**N. Users**	**Avg. Tweets**
Human	838	199
Bots	725	196

3.3 Simulations

Simulated Oracle. We simulated the oracle by training a logistic regression classifier with L2 regularization, with the default regularization C=1 parameter. The classifier was trained on the tweets in the train-split of the data. At every iteration the simulated oracle will return the predicted label of the queried tweet.

Student. For the student, we use a logistic regression classifier with the same configuration of the simulated oracle. We start every experiment with a small amount of labeled data, 50 user's timeline. T classifier is bootstrapped with the individual tweets of the initial 50 users.

4 Results and Discussion

4.1 Classification Methods

We tested several probabilistic classifiers and analyzed the learning curve by randomly sampling. Figure 4.5 show the accuracy learning curve of logistic regress with L1 and C=1, i.e., `lr-l1-c1`, L2 regularization with C=1, and C=10, i.e., `lr-l2-c1` and `lr-l2-c10`, and multinomial naive babes, i.e., `mnb`. We observe that `mnb` has the best performance followed by `lr-l2` with c=1 and c=10.

4.2 Data Representation

We set to find an appropriate data feature vector representation. We tested three data processing options: lower case, URL and mention collapse. We tried all eight combinations and tested using a five fold cross validation. In average, the accuracy of the classifier is 82.5% (+/− 0.4). We observed that there are no significant differences among all variations, we use the combination of all options active to reduce the size of the dictionary.

4.3 Other Features

We tested the use of other features derived from the text such uniqueness of the terms. For example, a bot may use the same terms over and over thus the number unique terms will be low. The uniqueness formulations was:

$$Unique(\mathbf{x}_i) = \frac{TF}{UniqueTerms} \tag{5}$$

where TF is the term frequency. We tested this feature representation and obtained 54% accuracy

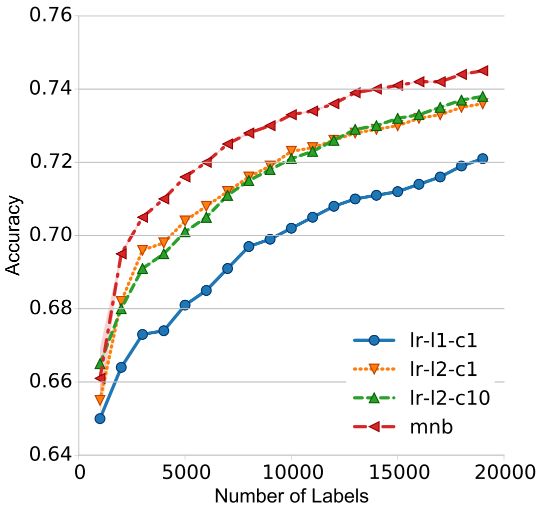


Figure 1: Classification models induced on tweets. The performance of four probabilistic classifiers on tweeter data. The budget is the number of labels.

(+/-0.0). We conjecture that the difference of this measure between the two classes is not significant to have enough classification power.

4.4 Error Analysis

We analyzed a fully trained classifier and reviewed the mistakes made on a test split. The idea is to identify if the classifier is able to reasonable annotate individual tweets, from a human point of view. We used a `lr-l2-c1` and we verify the incorrect labels with high incorrect probability. We found that the classifier usually mislabels tweets in other languages other than english, due to low number of examples. Common mistakes, also occurred on tweets with only mentions and URLs. Table 4.4 show examples of mistakes made by the classifier.

4.5 Active Learning Experiments

We tested several method of active learning where the learner algorithm picks a tweet from the timeline of a user and requests the label. We tried several approaches as follows:

`unc_first1` is a joint formulation of uncertainty of the timeline and maximum first tweet approach. This is a baseline.

`unc_rnd` is a joint formulation of uncertainty of the timeline and a random tweet. This is a baseline

`unc_sr` is a joint formulation of uncertainty and the best tweet. We tested using a logistic regression model and a multinomial naive bayes model.

Figure 2 shows the results accuracy results of the tested models. We observe that the multinomial naive bayes approach has the worst performance. `unc_first1` and `unc_rnd` are the close in performance, however, selecting random tweets perform slightly better in early iterations. We argue that the oracle responses are imbalanced thus introducing noise in the training data. Table 3 shows the confusion matrix of the oracle after labeling 690 tweets selected by the learning algorithm. Note that `unc_first1` and `unc_rnd` have a balanced percentage of error for both human and bot labels, whereas `unc_sr` mistakenly classifies more tweets as bots.

4.6 Other models and Bootstrap Effect

An important element of the active learning strategy is the bootstrap size of the training data. Bootstrap size can affect how good are the decision of the learner at early iterations. We tested sizes $bt \in \{10, 50, 100\}$ using STRUCTURED READING approaches and a logistic regression with L2 regularizations. Figure 3(a) shows the learning curve of the methods with different bootstrap sizes. We observe that the differences are not significant (note the standard error per method, i.e., shadowed in the graph),

Tweet Example	True Label	Observation
THIS_IS_A_MENTION THIS_IS_A_MENTION otro ,**que** a lo mejor no sabe que es la udef	Bot	Other languages
THIS_IS_A_MENTION ?effective **sales** and **marketing** is getting to the truth as quickly as possible? #sales #marketing #cmworl	Human	Terms used
rt THIS_IS_A_MENTION #goroyals	Bot	Only mention

Table 2: Example tweets incorrectly labeled by a lr-l2l-c1 classifier. The terms in bold face correspond to the highest weight value according to the classifier.

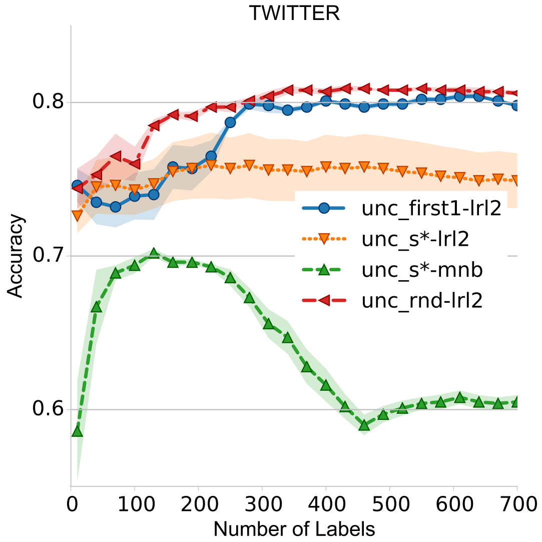


Figure 2: Active learning experiments. Results averaged over five trial.

unc_first1-lr

T. Size		H	B
690	**H**	40	12
	B	12	36

unc_sr-lr

T. Size		H	B
690	**H**	35	19
	B	8	39

unc_sr-mnb

T. Size		H	B
690	**H**	31	23
	B	8	38

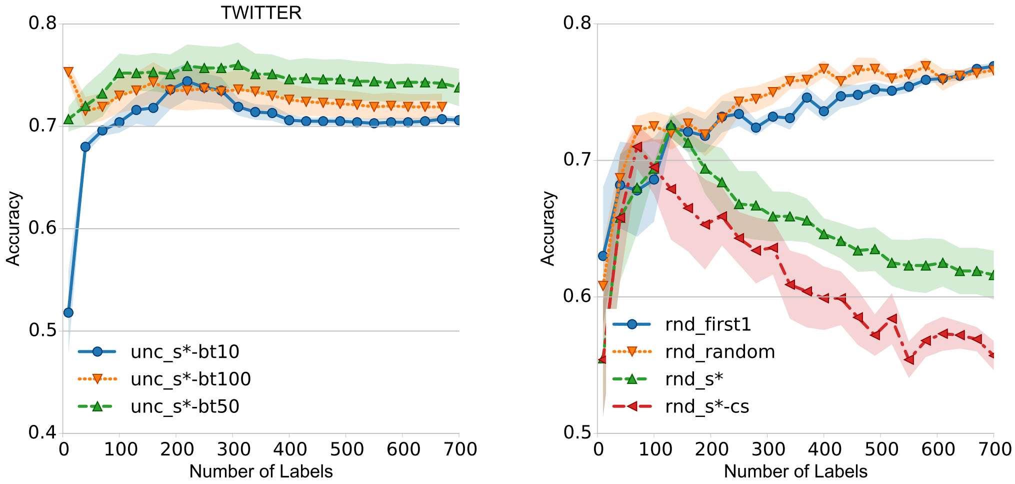
unc_rnd

T. Size		H	B
690	**H**	40	10
	B	11	39

Table 3: Oracle confusion matrix for different calibration methods on the Twitter dataset. The matrix is given in percentage with respected to the training size (**T.Size**). **H** are human labels and **B** are bot labels. Each row are true labels and columns are predicted by the classifier.

however, the smallest bootstrap is worse than using 100 user timelines.

We also wanted to test if a fixed penalty for the logistic regression model can improve the performance of our proposed method. Figure 3(b) shows the effect of changing the penalty of the model as the training data also changes. We used a random sampling method to isolate the effect of the penalty on the tweet selecting. We observed that the effect of the change does not improve the STRUCTURED READING methods to be comparable with the random and first1 baseline.



(a) Bootstrap Effect

(b) Adaptive Penalty

Figure 3: Effect of penalty and bootstrap on the STRUCTURED READING methods

5 Related Work

The current work on social bot detection typically uses network analysis techniques to construct complex features and induce a probabilistic classifier. Lee et al. (2011) proposed a classifier trained on a variety features such as demographics, friendship network, user content, and user history. **?** also proposes the use of alternative natural language processing features from the user's profile. In contrast to our method, we propose to use only the user content under an active learning framework.

Furthermore, he described methods on previous studies rely on available datasets, which are expensive to build. To the best of our knowledge, our work is the first attempt to use active learning to build a social bot classifier. However, our work is the first attempt to use an anytime active learning approach to address the task.

6 Conclusions

We proposed an active learning method to detect bot operated twitter accounts using a tweet from the users timeline. Identifying the automated nature of an account is a hard problem. We discuss possible alternative to build a training set of bot and human account by using active learning methods. We found that active learning methods can potentially save on annotation cost, however, further reducing the annotation cost by applying anytime active learning methods

References

Kyumin Lee, BD Eoff, and James Caverlee. Seven Months with the Devils: A Long-Term Study of Content Polluters on Twitter. *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

Maria E. Ramirez-Loaiza, Aron Culotta, and Mustafa Bilgic. Anytime Active Learning. In *AAAI Conference on Artificial Intelligence*, 2014.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *International Conference on Machine Learning*, pages 441–448, 2001.