International Journal of Research in Engineering, Science and Management
Volume-2, Issue-12, December-2019
www.ijresm.com | ISSN (Online): 2581-5792

Hateful Speech Detection on Social Media using Deep Learning: An Overview

Priya Jamdade[1], Shalini Manikrao Kudke[2], Anuradha Kale[3], Karuna Kamble[4], Ganesh Kalyani[5]
[1]*Professor, Department of Computer Engineering, Genba Sopanrao Moze College of Engineering, Pune, India*
[2,3,4,5]*Student, Department of Computer Engineering, Genba Sopanrao Moze College of Engineering, Pune, India*

***Abstract*: Hateful speech detection on Twitter is critical for applications like controversial event extraction, building AI chatter bots, content recommendation, and sentiment analysis. We are going to define that this task is able to classify a tweet as racist, sexist or neither. This task is being very challenging task due to the complexity of the Natural Language constructs. The proposed system carried out the text processing using supervised learning approach for Hateful speech detection in desired tweets. System also use polarity dataset for identify sentiment basis. The proposed system used deep learning approach for classification. We perform extensive experiments with multiple deep learning architectures to learn semantic word embeddings to handle this complexity.**

***Keywords*: Speech Detection, Deep Learning**

1. Introduction

Social media platforms (such as Twitter, Facebook, LinkedIn, Instagram) are one of the crucial means for communication and information dissemination over the internet.

Much can be learned about people's habitat by analyzing their behavior over the social media. This helps offenders to commit various cyber-crimes such as cyber bullying, skewing perceptions, misdirecting users to malicious websites, fraud, identity impersonation, dissemination of pornography, terrorist propaganda, to spread malware etc. Since identity deception provides means for offenders to commit such crimes it has become necessary to identify the fake identities over social media platforms.

These fake identities can be created by bots or humans. The fake identities by bots generally target large group of peoples at a time. Whereas, fake accounts by humans generally target specific individual or limited number of peoples. This system represents an approach detect the fake identities created by humans on social media platforms. In order to detect identity deception, we have applied Random Forest algorithm for machine learning. Also various preprocessing steps such as stop word removal, Porter's algorithm for stemming lexical analysis are applied on the data extracted through Twitter API. Accounts for bots are removed during data cleaning phase of preprocessing based on certain parameters such as presence of profile image, name etc., accounts of known celebrities are also removed from the corpus. The fake accounts are created using two random human data generator APIs and validated based on tests such as Mann Whitney U Test and Chi Square Test.

2. Literature Survey

The classification in machine learning is based on the training or learning from a training dataset. This learning is categorized into three types: supervised, semi supervised and unsupervised learning. In supervised learning class labeled data is present at the beginning. In semi supervised learning some of the class labels are known. Whereas; in unsupervised learning class labels are not available. As the training phase is finishes the features are extracted from the data based on term frequency and afterwards the classification technique is applied.

Estee et. al. [1] trained the classifier by applying previously used features in order to identify fake accounts created by human on Twitter. The training is based on supervised learning. They have tested for 3 different classifiers i.e. Support Vector Machine (SVM) with linear kernel, Random Forest (RF) and Adaboost. For SVM, the SVM Linear library in R software is used. Here the boundary based on feature vectors is created for classification. For RF model, the RF library in R software is used. RF model creates variations of trees and mode of class outcome is used to predict identity deception. For boosting model, the Adaboost function in R is used. Adaboost is used along with decision trees where each feature is assigned different weight to predict outcome. These weights are iteratively adjusted and output is evaluated for effectiveness of identity deception prediction at the iterations. This process is repeated until best outcome is obtained. Among these 3 classifiers the best result is chosen by the RF.

Sen et. al. [2] performed supervised learning based on features obtained from Fake Like data and Rand Like_data. They have experimented with different classification algorithms such as Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM) with RBF kernel, AdaBoost with Random Forest as base initiator; XGBoost and simple feed forward neural network i.e. Multi-Layer Perceptron (MLP) for detecting the fake likes on instagram. For MLP they have used 2 hidden layers with 200 neurons each. Both layers use sigmoid activation function and output layer has a dropout of 0.2 in order to prevent over fitting. Here, MLP outperforms other methods.

Sedhai et. al. [3] trained three different classifiers i.e. Naïve

Bayes (NB), Logistic Regression (LR) and Random Forest (RF) using semi supervised Learning. These three classifiers use different classification techniques i.e. generative, discriminative and decision tree based classification models. The dataset used was from Twitter. Twitter Id is detected as spam if at least two classifiers of these three detect it as spam otherwise it is detected as ham. They have called this framework as S3D (Semi Supervised Spam Detection). It gives best result as compared to any other individual classifier.

Xiao et. al. [4] performed supervised learning in order to extract best feature set from the LinkedIn data. They have trained three classifiers i.e. Logistic Regression (LR) with L1 regularization, Support Vector Machine (SVM) with radial basis kernel function and a Random Forest (RF) a nonlinear tree based ensemble learning method. Except regularization LR tries to find parameters using maximum likelihood criterion. Whereas with regularization there is tradeoff between fitting and having fewer variables to be chosen in the model. In this paper, they use L1 penalization to regularize the LR model. This technique maximizes the probability distribution of the class label y given a feature vector x and also reduces number of irrelevant features by using penalty term to bound the coefficients in L1 norm. The SVM looks for an optimal hyper plane as a decision function in high dimensional plane. While RF combines many weak classifiers (decision trees) to form strong classifier. For each decision tree training data is sampled and replaced to get training data of same size. Then at each node m features are selected at random to split decision tree. The common output class is considered as result of RF. Here RF gives the best result for classification of fake identities.

Ikram et. al. [5] used supervised two class SVM classifier implemented using scikit learn (an open source machine learning library for python) in order to automatically distinguish between like farm users from normal (baseline) users. They have compared this classifier with other well-known supervised classifiers such as Decision tree, AdaBoost, K- Nearest Neighbor (KNN), Random Forest (RF) and confirmed that two class SVM is best in detecting like farm users on Facebook.

Dickerson et. al. [6] used Indian Election Dataset (IEDS) extracted from twitter for training. They tried for six high level classifiers such as SVM, Gaussian naïve Bayes, AdaBoost, Gradient Boosting, RF and Extremely Randomized Trees. The classifiers were built and trained on top of scikit-learn, a machine learning toolkit supported by INRIA and Google. Here, AdaBoost performed best on the reduced feature set and gradient boosting performed best on full feature set where reduced feature set involved only those features that did not involve sentiment analysis.

Ikram et. al. [7] System used supervised two class SVM classifier implemented using scikit learn (an open source machine learning library for python) in order to automatically distinguish between like farm users from normal (baseline) users. They have compared this classifier with other well-known supervised classifiers such as Decision tree, AdaBoost, K- Nearest Neighbor (KNN), Random Forest (RF) and confirmed that two class SVM is best in detecting like farm users on Facebook.

Peddinti et. al. [8] developed a classifier that converts the four class classification problem into two binary classification problems: one that classifies each account as anonymous or non-anonymous and other classifies each account as identifiable or non-identifiable. The results of two classifiers are combined to classify each account as ‘anonymous’, ‘identifiable’ or ‘unknown’ for Twitter data. Both the binary classifiers use Random Forest (RF) with 100 trees as a base classifier. The choice of the classifier and number of trees is based on cross validation performance and out of bag error. These classifiers are also cost sensitive meta classifiers, where higher cost is imposed for misclassifying instances as anonymous or identifiable. The dataset used here was from Twitter.

Oentaryo et. al. [9] used supervised and unsupervised learning methods and tested for four prominent classifiers: naïve Bayes (NB), Random Forest (RF) and two instances of generalized linear model i.e. Support Vector Machine (SVM) and Logistic Regression (LR). This study involves Twitter dataset generated by users in Singapore and collected from 1 January to 30 April 2014 via the Twitter REST and streaming API. Here LR outperforms the other techniques and gives best result for classification of accounts as Broadcast bots, Consumption Bot, Spam Bot and Human.

Viswanath et. al. [10] uses unsupervised machine learning approach for training. The dataset used is from Facebook. They use K-Nearest Neighbors technique for this classification. In KNN data is classified based on majority vote of its neighbors, with test data being assigned to the class most common among its k nearest neighbors where k is a positive integer typically small in value. The classification is done into the four classes i.e. Black market, Compromised, Colluding, and Unclassified.

3. Objectives

- Implement a system with synthetic as well as real time text data which taken from any third party web applications.
- Implement trainings as well as testing phase for classification using sentiment base symbolic analysis like comment is happy, sad, excited, positive, negative etc.
- Successfully implement a Recurrent Neural Networks (RNNs) or FGA for detect the fake identities.
- Implement a Decision Tree (DT) for label classification.
- Evaluate the system with multiple experiments on different type data and analyze the accuracy as well as false ratio.

4. System Architecture

Proposed system provides Hateful speech detection in live streaming twitter data. A common characteristic of communication on online social networks is that it happens via

short messages, often using nonstandard language variations. These characteristics make this type of text a challenging text genre for natural language processing. Moreover, in these digital communities it is easy to provide a false name, age, gender and location in order to hide one's true identity, providing criminals such as pedophiles with new possibilities to groom their victims. It would therefore be useful if user profiles can be checked on the basis of text analysis, and false profiles flagged for monitoring. This research work presents an exploratory study in which system apply age group categorization approach base on the text features. The Recurrent Neural Networks (RNNs) has used for classification purpose. Finally, Decision Tree (DT) used for label creations.

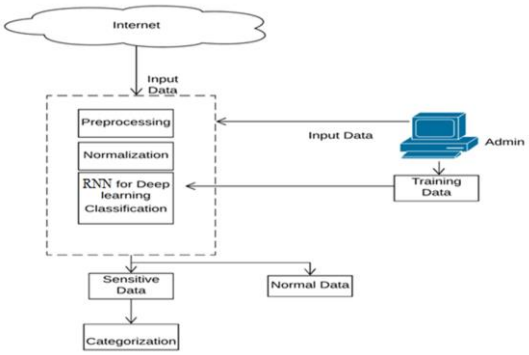

Fig. 1. Proposed system architecture

5. Methodology

Used Twitter API for training as well testing.

- Then we will apply various preprocessing steps such as lexical analysis, stop word removal, stemming (Porters algorithm), index term selection and data cleaning in order to make our dataset proper. During data cleaning step Bots are removed from the dataset based on certain parameters such as presence of name, profile image, number of followers, number of tweets etc. Also accounts of the known celebrities are removed from the given corpus.
- This was done to make the research results as realistic as possible. Most importantly, the following two statistical tests were employed to validate that the injected deceptive accounts are still representative of original mined corpus. Then supervised machine learning is applied in order to train the classifier. Here class labeled data is present at the beginning.
- Sentiment Analysis polarity algorithm for machine learning is applied to determine identity deception on social networks where multiple decision trees are created using randomly selected features from the feature set and the majority output class of all the decision trees is taken as output of the Random Forest.
- A system by learning representations through the user's age group classification use of Recurrent Neural Networks (RNNs) base on text terms, a significant increase in performance can be obtained on these tasks.

A. *Algorithms used*

Algorithms

1. Stop word Removal Approach
   Input: Stop words list L[], String Data D for remove the stop words.
   Output: Verified data D with removal all stop words.
   Step 1: Initialize the data string S[].
   Step 2: initialize a=0,k=0
   Step 3: for each(read a to L)
   If(a.equals(L[i]))
   Then Remove S[k]
   End for
   Step 4: add S to D.
   Step 5: End Procedure

B. *Stemming Algorithm*

Input: Word w
Output: w with removing past participles as well.
Step 1: Initialize w
Step 2: Intialize all steps of Porter stemmer
Step 3: for each (Char ch from w)
If (ch.count==w.length()) && (ch.equals(e))
Remove chfrom(w)
Step 4: if(ch.endswith(ed))
Remove 'ed' from(w)
Step 5: k=w.length()
If (k (char) to k-3 .equals(tion))
Replace w with te.
Step 6: end procedure

C. *TF-IDF*

Input: Each word from vector as Term T, All vectors V[i…n]
Output: TF-IDF weight for each T
Step 1: Vector = {c1, c2, c3….cn}
Step 2: Aspects available in each comment
Step 3: D = {cmt1, cmt2, cmt3, cmtn}
and comments available in each document
Calculate the Tf score as
Step 4: tf (t,d) = (t,d)
t=specific term
d= specific document in a term is to be found.
Step 5 : idf = t → sum(d)
Step 6: Return tf *idf

D. *Recurrent Neural Network*

Input: Training Rules Tr[], Test Instances Ts[], Threshold T.
Output: Weight w=0.0
Step 1: Read each test instance from (TsInstnace from Ts)
Step 2 :TsIns $= \sum_{k=0}^{n} \{Ak \dots An\}$
Step 3: Read each train instance from (TrInstnace from Tr)
Step 4 :$TrIns = \sum_{j=0}^{n} \{Aj \dots \dots Am\}$
Step 5: *w = WeightCalc(TsIns, TrIns)*

Step 6: if (w >= T)

Step 7: Forward feed layer to input layer for feedback FeedLayer[] ← {Tsf,w}

Step 8: Optimized feed layer weight, Cweigt←FeedLayer[0]

Step 9: Return Cweight

6. Expected Outcome

In such approach, there are 2 phases one is training phase in this phase system can train system with the help of sorted dataset. And another is testing phase in which system can test and analyze data with the help of systems proposed mechanism. System categorizes testing data into happy, sad, excited, positive, negative etc. System gets twitter data with the help of tweeter API. It is run time data accessing from user account. System is using a Decision Tree (DT) algorithm for label classification and Recurrent Neural Networks (RNNs)for to detect the fake identities.

The proposed system outcomes are as below once testing data applied to system,

- Recurrent Neural Networks (RNNs)can be used to detect the fake identities.
- Label classification is done by Implement a similarity weight for training rules.
- Give analysis graph and result on different type data with analyzing the accuracy as well as false ratio with the help of confusion matrix.

7. Conclusion

From this survey we conclude that the problem of detecting identity deception on social media can be solved by using various machine learning techniques such as SVM, RF, LR, NB, MLP, RNN and so on. Among these techniques deep learning the best performance with accuracy of 87.11 %. Also, we notice that the performance of the system varies with classification technique and dataset used. RNN can be used to solve the problem of determining fake vs. real identities on social networks with accuracy of 87.11 percent. The performance of given system varies with the dataset used for it.

8. Future scope

Furthermore, accuracy can be increased in future by enhancing features set and testing for other classification techniques such as deep learning with different activation functions. The performance of system can be increased by using other techniques such as Deep Learning with different activation functions in future.

References

[1] Estee Van Der Walt and Jan Eloff, “Using Machine Learning to Detect Fake Identities: Bots vs Humans,” IEEE, 2018.
[2] Indira Senet. al. “Worth its Weight in Likes: Towards Detecting Fake Likes on Instagram,” ACM, 2018.
[3] SurendraSedhai and Aixin Sun, “Semi-Supervised Spam Detection in Twitter Stream,” IEEE , 2018.
[4] Cao Xiao, David Freeman and Theodore Hwa, “Detecting Clusters of Fake Accounts in Online Social Networks,” ACM 2015.
[5] Ikramet. al., “Combating Fraud in Online Social Networks: Detecting Stealthy Facebook Like Farms,” ARXIV, 2016.
[6] Dickerson, V. Kagan and V. Subhramanian “Using Sentiment to Detect Bots on Twitter: Are Humans more Opinionated than Bots?” IEEE, 2014.
[7] Ikramet. al., “Combating Fraud in Online Social Networks: Detecting Stealthy Facebook Like Farms,” ARXIV, 2016.
[8] S. Peddinti, K. Ross and J. Cappos “Mining Anonymity: Identifying Sensitive Accounts on Twitter,” ARXIV ,2016.
[9] R. Oentaryoet. al. “On Profiling Bots in Social Media,” ARXIV, 2016.
[10] B. Viswanathet. al. “Towards Detecting Anomalous User Behaviour in Online Social Networks,” USENIX, 2014.
[11] M. Yahyazadeh and M. Abadi, "BotOnus: An online unsupervised method for botnet detection," The ISC International Journal of Information Security, vol. 4, pp. 51-62, 2012.
[12] M. H. Arif, J. Li, M. Iqbal, and K. Liu, "Sentiment analysis and spam detection in short informal text using learning classifier systems," Soft Computing, pp. 1-11, 2017.
[13] Estee Van Der Walt and Jan Eloff, “Using Machine Learning to Detect Fake Identities: Bots vs Humans,” IEEE, 2018.
[14] Indira Senet. al. “Worth its Weight in Likes: Towards Detecting Fake Likes on Instagram,” ACM 2018.
[15] Surendra Sedhai and Aixin Sun, “Semi-Supervised Spam Detection in Twitter Stream,” IEEE 2018.