IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 1, Ver. IV (Jan.- Feb. 2018), PP 59-62
www.iosrjournals.org

Accuracy Improvement For NLP Classification To Identify Messageson Social Media

Praveen Asthana[1,]Dr. Geetanjali Amarawat[2], Awanit Kumar[3]
[1](Completed MCA from Mysore University in 2003, India)
[2](Head of Department Computer Science and Engineering, Madhav University, India)
[3](Research Scholar, Department of Computer Science and Engineering, Madhav University, India)

Abstract: *Terrorist groups like ISIS and Jihadist are spreading their propaganda online using various forms of social networks like Facebook, Twitter, and YouTube. Best way to stop these groups is to suspend accounts that spread propaganda to harm living being and hurt any religious place or monuments. This can be done by analysts manually read and analyze an enormous amount of information on social media. Our proposed methodology is to attempt an automatically detect radical content that is released by jihadist and Terrorist groups on Twitter. We are using a branch of Artificial Intelligence i.e. Machine learning to classify a tweet as radical or non-radical and generate results which indicate that an automated approach to aid analysts detecting radical content on social media.*

Keywords: *Twitter, YouTube, Facebook, Machine Learning, Text Classification, NLP,and Naïve Bayes.*

Date of Submission: 23-02-2018 Date of acceptance: 010-03-2018

I. Introduction

Social media likes Facebook, Twitters is not only used to communicate with family and friends but also to promote radical views. In many cases individuals and organizations uses social media to attract and gain attention of volunteers' and fighters fundraisers to specific causes. Jihadists, people participating in a jihad[4], have aggressively expanded their use of Twitter as well as other social media applications such as YouTube and Facebook. In 2015 around 90000 Twitter accounts are suspected to support extremist groups [5].Sitting in front of a computer and promoting radical views on social media can be a valuable contribution to promote extremist groups. One example of this is a media mujahedeen. Mujahedeen is the plural form of mujahidin and is used to describe someone involved in jihad [6].Since 2011, members of jihadist forums have issued media strategies that encourage the development of a media mujahedeen. Guides describing how to use social media platforms and lists of recommended accounts tofollow are released in various forums [7]. One such guide is a Twitter guideentitled "The Twitter Guide: The Most Important Jihadi Sites and Support for Jihad and the Mujahedeen on Twitter". This guide outlines reasons for using Twitter and states that Twitter is an important arena of the electronic front. The guide has identified 66 important jihadist accounts that users should follow.A lot of research has been done on tweets classification where tweets are classified into several classes as in where tweets are classified as having a negative, neutral or positive sentiment or like in where tweets are classified into categories such as News, Events, Opinions, Deals, and Private Messages[1]. Not as much research has focused on classification of text as being radical/terrorist related or not.

II. Problem Statement

Analyzing Tweets makes it simple to understand what jihadist think and how they promote their propaganda (positively or negatively) about a certain keyword. One might be curious about spreading ISIS agenda can easily tweet with hashtag to make it trending on twitter so that others can view and can get motivated with ISIS agendas. Or, maybe they want to better understand what the positive and negative topics regarding their motivation about Islamic state. If people are tweeting about this keyword, then Analyzing Tweets can help us to categorize that conversation.In this research our objectives are:

- Find Terrorist tweets and their account by tweet words.
- Find tweets with Hashtag (Example: #isis, #Isis).

III. Text Classification

Text classification is a method to classifying a document into a predefined category [8]. In our proposed system the document is a tweeter tweet and the category is a Boolean value indicating if the tweet contains radical content or not. As in every supervised machine learning task a dataset is needed. The text classification process includes read the documents, preprocess the text (this may include tokenizing the text, lemmatizing, deleting stop words), create feature vectors, select features, and create a model.

Data Collection: The first step in creating a model is to ensure that a proper dataset is collected. In a raw format a dataset can consist of documents, images, sound recording etc. The data used to construct a model is called the training data.

Cleaning Data: When data is collected it is usually not ``clean" due to several reasons:

- Noisy- containing errors or outliers
- Misspelled words
- Unwanted elements: quotes (retweets), strange symbols.

After cleaning data some preprocessing needs to be done. Depending on the situation this may include:

- Stemming (bringing a word to its base form)
- Removing stop words
- Splitting sentences into tokens

IV. NLP (Natural language Processing)

Multinomial Naive Bayesis a classic algorithm for text classification and natural language processing (NLP).

In machine learning, we use feature vectorwhich is an n-dimensional vector of numerical features which represent some objects inside data. Usually machine learning requires a numerical representation of feature vectors because this facilitates mathematical computation and statistical analysis. The instance is often represented by an n-dimensional feature vector x = (x1... Xn)Rn, where each dimension is called a feature. The length n of the feature vector is known as the dimensionality of the feature vector. Feature selection includes count of words presence of words,presence of punctuation marks, count of punctuation marks, time-based features like the hour when a post was published. There are several feature selections methods. In this work we haveused the information gain algorithm to analyze feature selections. Information gain tells us how important a given attribute of the feature vectors is [9].

$$IG(T, a) = H(T) - H(T \mid a)$$

where T is a set of training examples of form (x, y) = (x1, ...,xn, y), xi is the value of the ith attribute of example x and y is the corresponding class label. H(T) is the information entropy and is computed as

$$-\sum_{i}^{n} P(x_i) \log_b P(x_i)$$

Where X is a discrete random variable and P(X) is its probability.

V. Creating Datasets

We have proposed three different datasets. We call these datasets as TW (PRO, RAND and Acon). Where Tw stand twitter, pro stands for promotion and Rand is Random value, here Rand value is Random twits and Acon is Account which is active by a Terrorist or jihadist (user accounts). Let's suppose our dataset has RAND which consists of 2000 random tweets and were not related to ISIS.

In Aconwe are talking about those accounts that were talking about ISIS and Terrorism but also and some of them were even against it but none of them supportingit such accounts asstopisisforever, No2ISISofficial.The assumption that these accounts were notposting messages supporting ISIS was made based on the user name and manual verification. Accounts with such user names are most probable promoting messages against ISIS/Jihadi.

Our proposed work is to find a suitable dataset that we can use to train our algorithms. We have collected a set of tweets containing hashtags that were related to jihadists, and in particular ISIS, from the English language spectrum of pro-ISIS clusters on Twitter. All of the hashtags are used are listed as #IS,#ISLAMICSTATE,

#ILuvISIS, #Islamicstate.It was also the case that not all the tweets were written in English and all of the tweets were messages supporting ISIS. Some of the messages were not related to ISIS at all and they had no violent/radical content. They were inside the corpus because they contained some similar hashtags like #IS.

In NLP problem, we will use the words or terms "tokens" of the document in order to classify it on the appropriate class.Tokenization is often used in lexical analysis. It is the process that splits a text up into words, phrases, symbols or other elements. this elements are called tokens and they are usually used forfurther processing.Tokenization is important in text processing because it allows processingeach item separately. For example tokenization can be the sentence "I luv jihad, kill more people!" after tokenization will be transformedto token1: "I", token2: "luv",token3: "jihad",token4: ",",token5:"kill",token6: "more",token7: "people",token8: "!". The corpuswas tokenized by using and adding some additional transformations.

VI. Naïve Bayes

By using the "maximum a posteriori (MAP)" decision rule, we come up with the following classifier:

$$c_{map} = \arg\max_{c \in C} \left(P(c \mid d) \right) = \arg\max_{c \in C} \left(P(c) \prod_{1 \le k \le n_d} P(t_k \mid c) \right)$$

Where tk are the tokens (terms/words) of the document, C is the set of classes that is used in the classification, $P(c \mid d)$ the conditional probability of class c given document d, $P(c)$ the prior probability of class c and $P(t_k \mid c)$ the conditional probability of token tk given class c.

we must estimate the product of the probability of each word of the document given a particular class (likelihood), multiplied by the probability of the particular class (prior). After calculating the MAP for all the classes of given set, we will select the one with the highest probability.

To handle numbers with specific decimal point accuracy, calculating the product of the above probabilities will lead to float point underflow. To avoid this instead of maximizing the product of the probabilities we will maximize the sum of their logarithms:

$$c_{map} = \arg\max_{c \in C} \left(\log P(c) + \sum_{1 \le k \le n_d} \log P(t_k \mid c) \right)$$

We have chosenthe one with the highest log score that the logarithm function is monotonic; the decision of MAP remains the same.

The last problem that we should address is that if a particular feature/word does not appear in a particular class, then its conditional probability is equal to 0. If we use the first decision method (product of probabilities) the product becomes 0, while if we use the second (sum of their logarithms) the log(0) is undefined. To avoid this, we will use add-one or Laplace smoothing by adding 1 to each count, Where B is equal to the number of the terms contained in the vocabulary V.

$$P\ (t \mid c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'}) + B'}$$

•Lemmatization: Lemmatization is often used in computational linguistics problems. It is a process that determines the lemma of a word [35]. In English a word can have different inflected forms. For instance the word 'walk' can be used as 'walked', 'walking', 'walks'. The base form of those words is 'walk'. This base form of the words is called lemma. For example a text like "He was with us yesterday and now he is tired" after lemmatizing will transformed to "He be with we yesterday and now he be tired". For lemmatization the toolkit was used.

VII. Experimental Word

We have used NLTK (natural language toolkit) for breaking up sentences into words (tokenization): "IsIS will kill people" tokenizes to a list of individual words: "IsIS", "wil", "kill", "people".Reducing words to their stem (stemming): "will" stems to "wil" 0r "IsIs'll" which allows it to be matched with "will, will be" (same stem)Next step is to provide some training data. A few sentences that is associated with each intent (a "class"). If the user says "kill non jihadi", we want that to be the "greeting" intent and take it as negative tweet.

We will have to notice that this is a list [] or dictionaries {}, In R we don't have dictionary but in python we uses dictionary and inside key we define certain word inside the key . We have to notice the "corpus" (an NLP term) a collection of all stemmed words. The stem of "kill" is "kil" so that it matches the stems for "killing", as a example.

We have now organized our data into 2 dictionaries: **corpus_words** (each stemmed word and the # of occurrences) **class_words** (each class and the list of stemmed words within it). Our algorithm will use these data structures to do its thing.For our experiments we used Naive Bayes classifierfor the default configuration.

Our next step is to write an algorithm, our input data is transformed consistently with our training data.Let's try classifying the sentence "ISIS will attack on siria to kill USA-Army?"

match: will
match: siria
match: kill
Class: US-Army Score: 3
match: attack
match: kill
match: US-Army
Class: Siria Score: 3
match: ISIS
match: will
Class: Attack Score: 2

Each class generates a total score for the # of words that match. our algorithm improves by accounting the commonality of each word. The word "Kill" should carry a lower weigh than the word "Will" in most cases, because it is more common.Our last step is to abstract the algorithm so it can be used simply. And now we can test our classifier with inputs.

Wekatool is useful in setting each word separately but it builds up a representation of whole sentences based on the sentence structure. The sentence is computed based on how words compose the meaning of the sentence. The values can take as: very negative, negative, neutral, positive, very positive.

Parts of speech (POS): Opinion words are commonly used to express opinions including good or bad, like or hate. Feature selection methods Feature Selection methods can be divided into lexicon-based methods that need human annotation, and statistical methods which are automatic methods that are more frequently used. Lexicon-based approaches usually begin with a small set of 'seeds' words. Then they bootstrap this set through synonym detection or on-line resources to obtain a larger lexicon. This proved to have many difficulties. Statistical approaches, on the other hand, are fully automatic. The feature selection techniques treat the documents either as group of words (Bag of Words (BOWs)), or as a string.

VIII. Result

We performed experiments using different classification techniques,datasets, and features to evaluate the ability these techniques have to distinguishextremist from non-extremist tweets. We also examine which featuresappeared to be most important in facilitating this task. The Data collection was of approx. Between 100 and 150 tweets were used in total.Results differed depending on what datasets we used. Using TWPROand TW-RAND led to better results than if TW-PRO and TW-CONwere used. The results for TW-PRO and TW-RAND and the features (S +

T + SB) can be noted performs very well with 100 % accuracy on the test set.

References

[1]. http://ieeexplore.ieee.org/document/7842262/
[2]. https://www.sciencedirect.com/science/article/pii/S2090447914000550
[3]. http://uu.diva-portal.org/smash/get/diva2:846343/FULLTEXT01.pdf
[4]. Devin R Springer. Islamic radicalism and global jihad. Georgetown University Press, 2009.
[5]. Security Council Adopts Resolution 2170. http://www.un.org/press/en/2014/sc11520.doc.htm, 2014. [Online; accessed 20-June-2015].
[6]. Mujahideen. http://terrorism.about.com/od/m/g/Mujahideen.htm. [Online; accessed 10-June-2015].
[7]. NicoPrucha Ali Fisher. The call-up: The roots of a resilient and persistent jihadist presence on twitter ctx. 4(3). August 2004.
[8]. Thorsten Joachims. Text categorization with support vector machines:Learning with many relevant features. Springer, 1998.
[9]. Sentiment analysis. http://nlp.stanford.edu:8080/sentiment/rntnDemo.html. [Online; accessed 30-April-2015].
[10]. http://blog.datumbox.com/machine-learning-tutorial-the-naive-bayes-text-classifier/