Tracking and Quanifying Negative Content on Social Media

Manish Ranjan
Computer Science
University of Georgia
Athens, USA
ranjan@cs.uga.edu

Shannon Quinn
Computer Science
University of Georgia
Athens, USA
spq@uga.edu

***Abstract*—In this paper, we propose an unsupervised learning approach to track and detect negative content on social media. Our approach exploits topic modeling to generate feature vectors for individual users based on their topic content contribution at Twitter and cluster them. Out of randomly selected 6000 users we filter top 5% based on their negative feature dominance and attain at least 95.5% accuracy in verifying them using sentiment analysis and manual validation approach, both independently. Our framework uses t-SNE (t-Distributed Stochastic Neighborhood Embedding) to find co-emerging negative topics in a dynamic way. This approach gives us an edge over existing frameworks by detecting unobserved negative topics like any terrorist-attack, virus, health hazard, etc. by going totally unsupervised learning way. Hence, our approach provides an alternative to designing large scale hand coded frameworks to detect changing events on social media, such as zika or ebola, and discover such keywords dynamically.**

Keywords—Twitter, topic modelling, machine learning, sentiment analysis, clustering

I. Introduction

Social media is actively used by 68 % of total 3.14 billion active internet user's family [1]. Hence, we could say that social media sites are most popular on internet. With over 200 million active users and almost half a billion tweets per day [2], [3] Twitter has been one of the most popular social media for real time information sharing. However, negative content included in these user generated data, could sometime offer a great source of information trend (health hazard outbreaks like zika, ebola) and sometime unwanted or undesirable data (in terms of adult content). Such undesirable data can make users experience on website unpleasant and may also be something that certain users want to filter (e.g. parents). An example of this is illustrated in Figure 1. Twitter in past few years has also become one of the most used platform for the adult entertainment industry to conduct social marketing campaigns. [15] [16]. As for example, the user marked in Figure1 shares adult content frequently, yet appears on search.

In this paper we wanted to take a generic approach to detect such negative content as well as users who repeatedly shares them. This is difficult problem as large numbers of account gets created on Twitter every day to promote services related with adult industry. Because of this the complain of adult content spread has already started appearing [17] [18]. Hence, we also wanted the system to be scalable and work at scale of twitter while detecting such users so it can keep up with spammers.

For a given twitter stream snapshot input we quantify and project users on a scale of 0-1 based on their shared negative content, 0 being not sharing negative content at all and 1 being sharing extremely negative content. We also wanted to look at topics which were co-emerging with negative topics and bring a sense of distance concept between such topics to measure the correlation accurately. e.g. adult content's topic distance with download and food. As a general rule we could expect that topic distance of 'adult content' will be lesser with electronics (mobile, download, etc.) than with food. In this paper we opted to focus on adult content as example of negative content, given that this could be one of the most common inappropriate content on social media. However, designed platform is generic enough to handle any negative topic.

In our framework, we consume twitter stream without any filter for a predetermined time interval and then clean the data for only text. This paper focuses only on text data analysis. Sentiment analysis is used to verify the top 5% most negative users. As a third and most safe measure, we also handpicked users to verify the accuracy of at least 95.5%. We also later attempt to generalize it by labeling user based on Algorithm 2 and 3 and still achieve at least 95% accuracy.

The major contribution is of three fold.

1. Using LDA (Latent Dirichlet allocation) algorithm to generate feature vector of users based on their personal tweet content similarity with corpus topic. To the best of our knowledge, our work in this paper is first such attempt. A representation of such vector is shown in Figure 5.

2. Our Approach in this paper is to focus on designing, Generic pipeline to detect user interest on a scale of [0,1] for any topic appearing in corpus, is first attempt in this direction.

3. We also introduce for the first time, the concept of topic distance. e.g. Distance of food with politics, adult content with electronics. This could be really helpful while working at scale of twitter to find topic of interest automatically.

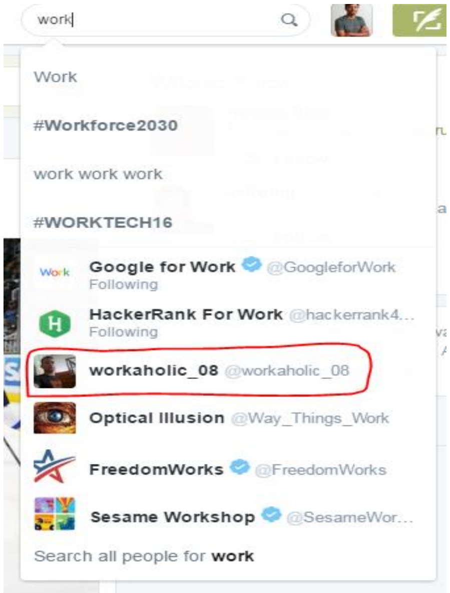

work
Work
#Workforce2030
work work work
#WORKTECH16
Google for Work @GoogleforWork
Following
HackerRank For Work @hackerrank4...
Following
workaholic_08 @workaholic_08
Optical Illusion @Way_Things_Work
FreedomWorks @FreedomWorks
Sesame Workshop @SesameWor...
Search all people for work

Figure 1: Example of how regular search string like work can bring a user like marked in red who shares explicit adult content on Twitter very frequently.

II. BACKGROUND

This section reviews the efforts made in the direction of detecting negative content on social media. However, to the best of our knowledge, we have not found any dedicated work toward finding negative co-emerging topics and detecting users who share negative content on quantitative scale. However, there is a latest work by Hanquiang Cheng et al,2015 specifically on designing a social classifier to detect Adult accounts on Twitter [4]. We will talk about its limitation in Section D.

A. *Detecting negative content*

Detecting negative content is a very broad criterion. Negative could mean spam, could mean talking about disease, violence, and even about health hazard in some cases. Some previous work with impressive results are "Detecting Offensive Tweets via Topical Feature Discovery Over a Large Scale Twitter Corpus" [5], "Uncovering Social Spammers: Social Honeypots + Machine Learning" [6], "Detecting Spammer on Social Network" [7], "Predicting Depression via Social media" [8] etc. Topic Modeling approach for spam detection for reviews was explored in detail by Jiwei Li et.al, [6]. However, none of the research so far have directly explored the topic modeling usage. Most of the adult content detection has really focused approach of detecting it via image processing techniques. However, to the best of our knowledge we have not seen any dedicated work toward find topics which co-emerge on social media with negative topics and finding relative topic distance concept from a given topic.

B. *Sentiment analysis of negative content on twitter*

Sentiment analysis on Twitter has drawn significant attention and enormous amount of work related to sentiment analysis of tweets has already been published, such as [9], [10], [11]. Detection of negative sentiment across tweet corpus is rather straight forward.

C. *Bootstrapping negative dictionary for sentiment algorithm*

Bootstrapping a dictionary with negative words with relevant score has achieved high accuracy for detecting sentiment across tweets using bag of word model. For most of such relevant work we would like readers to refer De Choudhury 2013 [8]. Our Dictionary sources are AFINN and Opinion Mining, Sentiment Analysis, and Opinion Spam Detection. [12], [13]

D. *Latent Dirichlet Allocation on twitter data for topic detection*

LDA since its inception, has been used on various text corpus to detect topics in supervised manner. Usage of LDA on twitter corpus has given insight on community detection [14], Crime detection [26], health related communities and discussions [27].

Iterative social based classifier for adult content detection,2015 [4] is one latest attempt to address this problem. However, the method can only detect adult Twitter account accurately using a small number of labeled account. Hence it is semi supervised approach. We wanted to approach this problem from unsupervised angle so we can detect not only what is established as a negative content but also topic appearing in context of negative content for the very first time, dynamically, without any manual intervention. Hence a more flexible and dynamic approach without giving away on accuracy. Our generic approach gives us a lot of information on user's tweet correlation with every topic present in corpus, however we have chosen to take adult content as an example for further analysis.

E. *t–SNE on high dimension twitter data*

t-SNE as a visualization technique for high dimensional data has been used in many contexts [23], [24]. Using it to find negative topics and also finding topic distance of other topics from negative content topic (adult content in this case) is first such effort.

III. PIPELINE DESIGN OVERVIEW

The architecture of the system is given in Figure 3. We collected twitter data from stream end with just one filter of language set as English. This is how the initial user selection pipeline looked like.

A. *Twitter Corpus*

Our tweet corpus contains only the text contents from the filtered 10,000 users. Once we have the users DB, next we use Twitter api to download all the tweet content of these users.

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

Twitter allows us to download last 3200 tweet for a given userid. We take these 6000 users and download for each userid, last accessible tweets. Our training set had 18 million tweets and test set had 12 million tweets. While the data at Twitter is really unique as it reflects views, opinions of millions of user, it also offers few unique challenges.

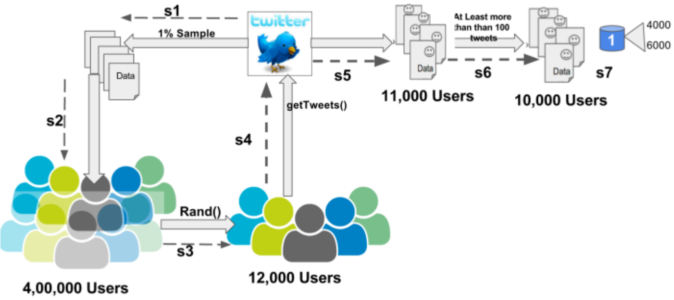


Figure 2: User selection pipeline for further analysis s1) connect to Twitter stream to collect 1% random tweets s2) filter users based on language setting as English s3) random function to select 12000 users s4) query Twitter for last available tweets of these 12000 users. s5) collect those tweets. S6) Filter users who has less than 100 tweets s7) DB: from these 10,000 users select 6000 for analysis and 4000 for validation.

Twitter as platform provides an environment very conducive to the rapid development of online rhetoric and events. These rhetorics neither appears in dictionary nor are formally recognized as a part of English language. But it carries semantic meaning to large number of users, therefore critical to identify user's intent. Also because the tweet's text is limited to 140 characters, users try also to place multiple words together with incomplete/incorrect spellings. This makes data processing and building models in NLP domain a challenge. As the problem is unsupervised in some sense because essentially we do not have a priori knowledge of all possible terms that are indicative of negative intention. A very relevant and latest example would be the hashtag #chickenTrump which appeared in largely negative context on twitter however word itself contains no meaning what so ever. Furthermore, these terms are changing constantly, as new terms appear on the fly in response to changing circumstances.

B. Twitter Preprocessing

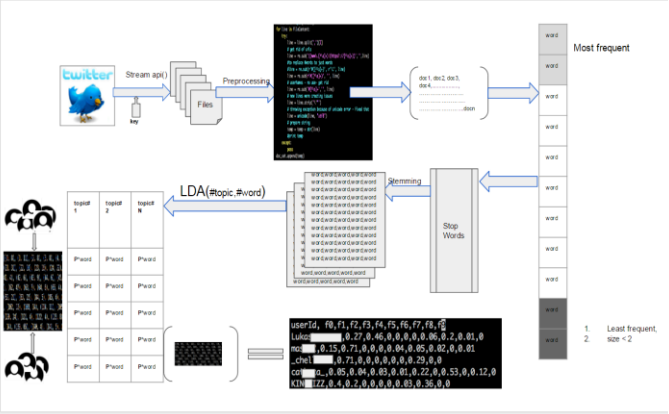
We designed a word cleaning pipeline, applying a series of filters on the input data to create a quality dictionary which will be input to topic modeling algorithm LDA.

1. Removed non-English tweets using. We used "guess_language" python package to do so [19].
2. We also created a HashSet[20] of text shared for each user to do away with repetitive sharing of certain text which can create bias in word frequency, hence impact LDA.
3. Removed all the links, shortened URLs.
4. Users on Twitter use @username format to refer to other people. We removed that.
5. Remove #words to words. The # symbol is called hashtag in twitter. Twitter users use it to create online rhetorics.
6. Converted all the words to smaller case to get rid of case error, like Love and love will become the same word.
7. stop-words [29] removed using NLTK package [30].
8. We got rid of all the words where word length is less than two.
9. We convert words like lover, loves, loving, etc. in to one single word love using stemming technique. [31]
10. We also got rid of all the words, where word frequency in the given document (a document here is one user's entire tweet text) is less than 4. We get to the number "4" based on number of experiment we ran and measured against quality of topic we were getting. Feature Engineering

The idea is to consider each user's tweet corpus as one document, after applying all the filters mentioned under previous Twitter pre-processing step. The result of our cleaning stage are texts which produce, stop-words filtered and stemmed list of words, from each individual document. So we can imagine looping through all our 6000 user's individual documents, and appending all of them to a single list, then what we have at the end of this processing step is a list of lists, one list for each of our original document containing all the words.

Next is to create a dictionary from these texts. The quality of dictionary is directly proportional to quality of topics LDA will be able to generate. Because twitter is known for all the misspelled and rhetorics, we also filtered the words via a Webster dictionary, so that obtained topics are more meaningful as far as sensible topics are concerned. This step considerably increased the processing time of algorithm, but was vital to getting the meaningful word distribution under each topic.

Once we generate document term matrix [21], we feed this as input to LDA model, along with no of topic as input. Later we used individual user's bow (bag of word) [23] to determine user's shared text correlation with corpus topic on a scale of 0-1 where 0 being not related at all and 1 being maximum correlation with that particular topic. We used genism [22] distributed package to implement LDA.



```
userId, f0,f1,f2,f3,f4,f5,f6,f7,f8,f9
Luka[redacted],0.27,0.46,0,0,0,0,0.06,0.2,0.01,0
mo[redacted],0.15,0.71,0,0,0,0.04,0.05,0.02,0,0.01
_chel[redacted],0.71,0,0,0,0,0,0,0.29,0,0
co[redacted]a_,0.05,0.04,0.03,0.01,0.22,0,0.53,0,0.12,0
KIN[redacted]IZZ,0.4,0.2,0,0,0,0,0.03,0.36,0,0
```

Figure 3: Pipeline's architecture to generate individual user's feature in correlation with corpus topic by taking number of topic equal to 10 as example. The pipeline starts from extracted tweets for all users and follows arrow direction until it generates user's feature

The output of LDA on tweet corpus looked like this.

$$T_1 = p_{11} * w_{11} + p_{12} * w_{12} + \dots + p_{1n} * w_{1n}$$
$$T_2 = p_{21} * w_{21} + p_{22} * w_{22} + \dots + p_{2n} * w_{2n}$$
$$.$$
$$T_n = p_{n1} * w_{n1} + p_{n2} * w_{n2} + \dots + p_{nn} * w_{nn}$$

$where$
$n = number of topics$
$p_{11} = probability\ of\ topic\ 1\ word\ 1$
$w_{11} = word\ 1\ of\ topic\ 1$

$trainedModel = [T_1 + T_2 + \dots + T_n]$

Figure 4: Topic modeling output and the relevant definitions, where T_1 is topic 1, $P_{1,1}$ is probability of word 1 in topic 1 and w_{11} is word 1 of topic 1.

Once we had the above traineddModel we ran following algorithm to get to the feature vector of each user.

```
Algorithm 1 Finding Feature Vector of Each User Based on Their Shared Tweets
procedure BUILDFEATUREVECTORFROMTRAINEDLDA (TRAINED-MODEL)
    for each topic 10, 20, 30, 40 and 50 do
        save text with filters mentioned in section 3.B
        dictionary = corpora.Dictionary(texts)
        for each text in texts do
            corpus = dictionary.documentToBagOfWord(text)
        end for
        for each item in corpus do
            print trainedModel[item]
        end for
    end for
end procedure
```

We used genism package [22] to implement the algorithm. Once the algorithm was run, the feature vector was generated where size of feature vector was equal to number of topics generated out of tweet corpus.

$user_x = [(T_0, 0.05172770755943485), (T_1, 0.02335369948698876 4), (T_2, 0), (T_3, 0.033210820415885$
$(T_4, 0.07116176864198309 2), (T_5, 0.63556554249704389), (T_6, 0.16274986525749854), (T_7, 0.019396170$
$(T_8, 0), (T_9, 0)]$

Figure 5 : FeatureVector's Data Structure example for 10 Topic for a random user. Such 6000 for trained and 4000 for test data feature vector was generated

C. Clustering

Next we wanted to run few unsupervised clustering algorithms to see how people who have maximum correlation with negative content, group together. We wanted to try two broader categories in clustering: soft clustering and hard clustering [28]. In hard clustering category, we went with K-Mean clustering and in soft clustering category we went with Expectation Maximization (EM). For topic size 10, 20, 30, 40 and 50 we ran both algorithms. We wanted to verify if using feature vectors, we will be able to group similar elements together. As expected, for lower dimension K-Mean worked as good as EM, but with dimensions increased to 50, K-Mean started suggesting a very high number of clusters as shown in Figure 6.

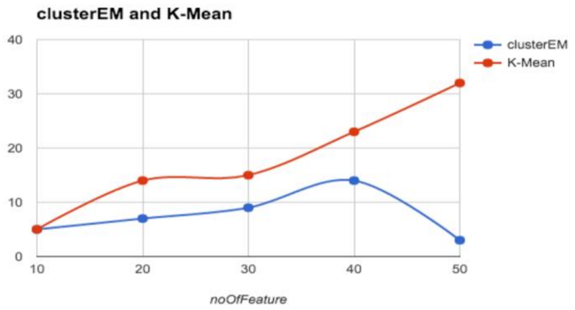


Figure 6: No of clusters suggested by hard clustering and soft clustering algorithms based on the feature vector length.

D. Assigning independent class to every instance of obtained feature vector for clustering of only "dominant" negative users

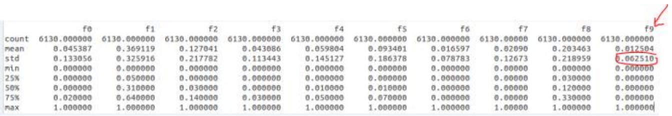
1. To assign class to every instance we need to analyze the feature vector sample shown in Figure 5. For simplicity, we will be taking only number of topic equal to 10 as example. Detailed comparative analysis among the topics will be explained in section 4-Experiments**.**

```
Algorithm 2 Assigning class to each row instance based on dominant feature
procedure ASSIGNCLASS(FEATUREVECFILE)
    for each line in featureVecFile do
        class = max($f_0, f_1, .....f_9$)    end for
```

AS an output of this Algorithm 2, an individual feature vector gets assigned a class based on max feature value.

2. Labeling N (Negative), NN (NonNegative) to find co-emerging negative topics.

To label instances with N/NN, we first found the mean, standard deviation and other statistical values of interest of feature values, independently.



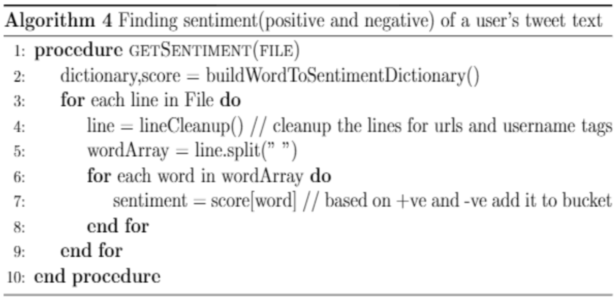
	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9
count	6130.000000	6130.000000	6130.000000	6130.000000	6130.000000	6130.000000	6130.000000	6130.00000	6130.000000	6130.000000
mean	0.045387	0.369119	0.127041	0.043086	0.059804	0.093401	0.016597	0.02090	0.203463	0.012504
std	0.133056	0.325916	0.217782	0.113443	0.145127	0.186378	0.078783	0.12673	0.218959	0.062510
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
25%	0.000000	0.050000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.030000	0.000000
50%	0.000000	0.310000	0.030000	0.000000	0.010000	0.010000	0.000000	0.00000	0.120000	0.000000
75%	0.020000	0.640000	0.140000	0.030000	0.050000	0.070000	0.000000	0.00000	0.330000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.00000	1.000000	1.000000

Figure 7: Mean, Standard deviation of 10 topic feature vector. The marked part explains the standard deviation see for negative topic

Once we had these values for each feature, looking at the Topic list generated from the corpus, we figured say if T_1 is the negative topic. Then following algorithm decides which instances get labeled as N or NN.

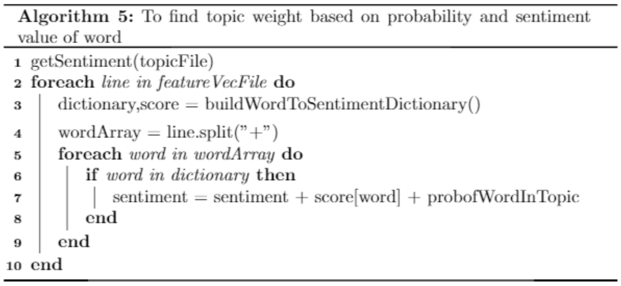
```
Algorithm 3: Assigning Binary class (N/NN) ro each row instance
    /* here std-10 is the value of
       $T_1 - Negative\ Topic\ standard\ deviation\ value$        */
    $val \leftarrow std - 10$
    assignClass(featureVecFile)
    foreach $line\ in\ featureVecFile$ do
        valOfFeature = line.valueAt($T_1$) // assuming T1 is negative topic
        if $valOfFeature \geq val$ then
            $class = "NN"$
        end
        else
            $class = "N"$
    end
```

E. *Sentiment analysis to cross verify the negative clusters accuracy*



```
Algorithm 4 Finding sentiment(positive and negative) of a user's tweet text
procedure getSentiment(file)
    dictionary,score = buildWordToSentimentDictionary()
    for each line in File do
        line = lineCleanup() // cleanup the lines for urls and username tags
        wordArray = line.split(" ")
        for each word in wordArray do
            sentiment = score[word] // based on +ve and -ve add it to bucket
        end for
    end for
end procedure
```

F. *To find topic weight based on sentiment score and probability of word*



```
Algorithm 5: To find topic weight based on probability and sentiment value of word
getSentiment(topicFile)
foreach line in featureVecFile do
    dictionary,score = buildWordToSentimentDictionary()
    wordArray = line.split("+")
    foreach word in wordArray do
        if word in dictionary then
            sentiment = sentiment + score[word] + probofWordInTopic
        end
    end
end
```

G. *t- SNE on high dimensional feature vector*

Post methods followed in section D.1, we had feature vectors with class assigned. For 10 topic feature vector, the class label will follow range from 0-9, 0-19 for 20 topic feature vector, 0-29 for 30 topics and so on. Following section D.2, we will get only two class labels, N and NN. t- SNE provides a 2 dimensional mapping of high dimensional image, while still keeping sense of distance among the feature vector. This was a very useful visualization technique to understand user distribution of negative content as well as the topics which co-emerge with negative content. e.g. for the data in experiment, we found topic containing videos, download, Trump (presidential candidate) being much closer to negative cluster than other topics. Figure 16 will explain this in detail.

IV. EXPERIMENTS

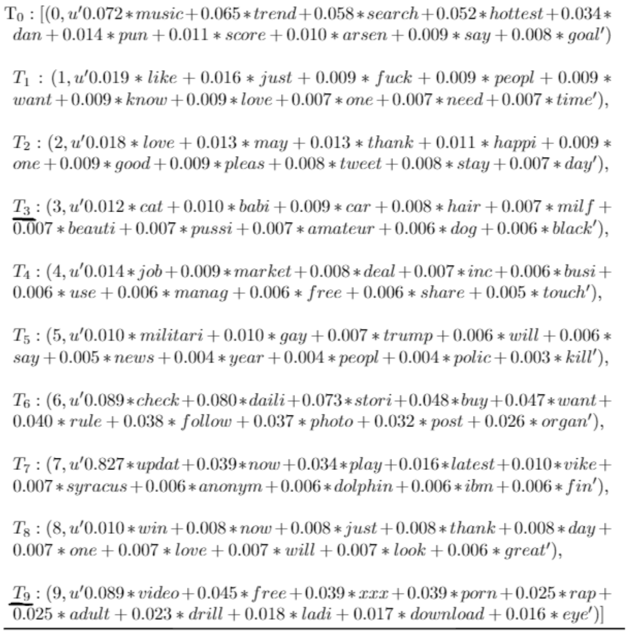
We had 15 million tweets from 6000 unique users. As a first step we used preprocessing steps mentioned in section III-B. Post that we used distributed genism package to run LDA on top of this text corpus. We ran it on topic 10, 20, 30, 40 and 50 as input to see topic distribution. We will be discussing experiments with 10 topics. The rest of charts, reports and code we intend to make public in near future. The 10 topics generated are as shown in Figure 8. The topic with most of the adult content is marked as T_9 and T_3.

Now we can use this trained LDA model to generate the feature of users whose text was used to generate the LDA model. A small sample of such generated feature is shown in Figure 9. Once we had these feature vectors, there were two types of users we were interested in detecting.

1. Users who share aggressively only negative content (Most of them turned out to be sharing adult content)
2. User who share negative content in context. E.g. Politics, Videos, conversation etc.

A. *Finding aggressive negative content sharing users*

Once we had generated feature vectors for all the 6000 users, we looked at the topics and T_9 was the most negative topic with all the negative words appearing. Hence we sorted the feature vector based on most dominant T_9 column value and picked top 75 users for a closer look, manually. Figure 10, is how the distribution looked like. The content of T_9 suggests that download and adult content appears in a document quite frequently on Twitter. Hence we can see that we had at least 68.4% accuracy in detecting users who share negative(adult) content aggressively.



$$\mathrm{T}_0 : [(0, u'0.072*music+0.065*trend+0.058*search+0.052*hottest+0.034*dan+0.014*pun+0.011*score+0.010*arsen+0.009*say+0.008*goal')$$

$$T_1 : (1, u'0.019*like+0.016*just+0.009*fuck+0.009*peopl+0.009*want+0.009*know+0.009*love+0.007*one+0.007*need+0.007*time'),$$

$$T_2 : (2, u'0.018*love+0.013*may+0.013*thank+0.011*happi+0.009*one+0.009*good+0.009*pleas+0.008*tweet+0.008*stay+0.007*day'),$$

$$T_3 : (3, u'0.012*cat+0.010*babi+0.009*car+0.008*hair+0.007*milf+0.007*beauti+0.007*pussi+0.007*amateur+0.006*dog+0.006*black'),$$

$$T_4 : (4, u'0.014*job+0.009*market+0.008*deal+0.007*inc+0.006*busi+0.006*use+0.006*manag+0.006*free+0.006*share+0.005*touch'),$$

$$T_5 : (5, u'0.010*militari+0.010*gay+0.007*trump+0.006*will+0.006*say+0.005*news+0.004*year+0.004*peopl+0.004*polic+0.003*kill'),$$

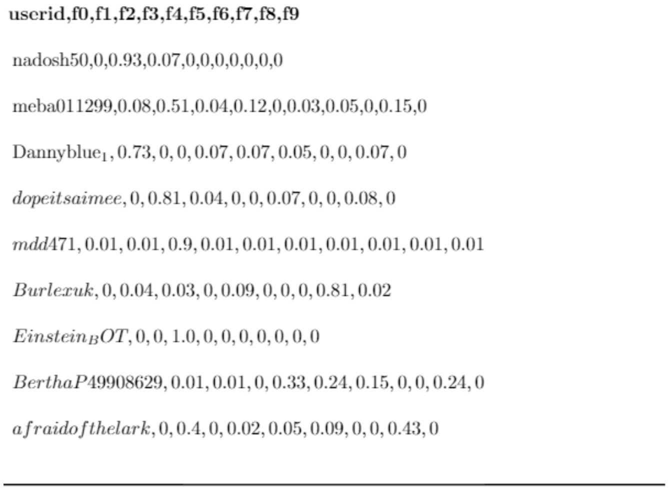
$$T_6 : (6, u'0.089*check+0.080*daili+0.073*stori+0.048*buy+0.047*want+0.040*rule+0.038*follow+0.037*photo+0.032*post+0.026*organ'),$$

$$T_7 : (7, u'0.827*updat+0.039*now+0.034*play+0.016*latest+0.010*vike+0.007*syracus+0.006*anonym+0.006*dolphin+0.006*ibm+0.006*fin'),$$

$$T_8 : (8, u'0.010*win+0.008*now+0.008*just+0.008*thank+0.008*day+0.007*one+0.007*love+0.007*will+0.007*look+0.006*great'),$$

$$T_9 : (9, u'0.089*video+0.045*free+0.039*xxx+0.039*porn+0.025*rap+0.025*adult+0.023*drill+0.018*ladi+0.017*download+0.016*eye')]$$

Figure 8: Topic generated by LDA from the tweet corpus



userid,f0,f1,f2,f3,f4,f5,f6,f7,f8,f9

nadosh50,0,0.93,0.07,0,0,0,0,0,0,0

meba011299,0.08,0.51,0.04,0.12,0,0.03,0.05,0,0.15,0

$\mathrm{Dannyblue}_1, 0.73, 0, 0, 0.07, 0.07, 0.05, 0, 0, 0.07, 0$

$dopeitsaimee, 0, 0.81, 0.04, 0, 0, 0.07, 0, 0, 0.08, 0$

$mdd471, 0.01, 0.01, 0.9, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01$

$Burlexuk, 0, 0.04, 0.03, 0, 0.09, 0, 0, 0, 0.81, 0.02$

$Einstein_BOT, 0, 0, 1.0, 0, 0, 0, 0, 0, 0, 0$

$BerthaP49908629, 0.01, 0.01, 0, 0.33, 0.24, 0.15, 0, 0, 0.24, 0$

$afraidofthelark, 0, 0.4, 0, 0.02, 0.05, 0.09, 0, 0, 0.43, 0$

Figure 9: Feature generated using trained LDA model

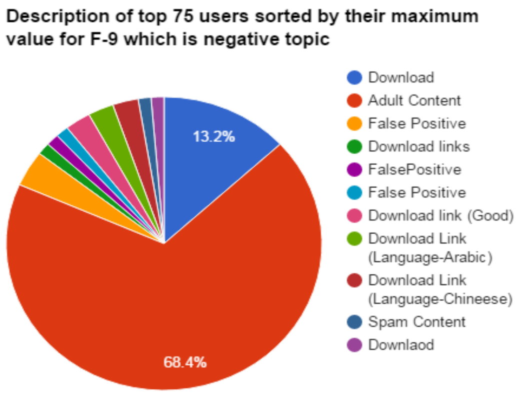


Figure 10 Top 75 user's distribution based on negative topic correlation

As a second technique, we used Sentiment analysis to see if were doing better than a regular sentiment analysis approach.

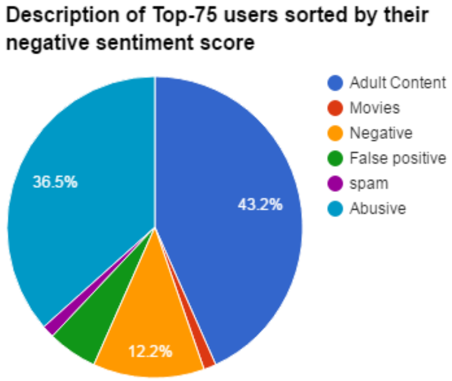


Figure 11 Top 75 user's distribution based on sentiment score

We took the same 6000 sample and ran sentiment analysis on top of it by implementing Algorithm 4. Among the top 75 users using sentiment analysis technique, we had at least 43.2% (Adult Content only) accuracy in detecting users who share adult content online.

As we can see from above two charts that we were more accurate in detecting users who exploit social media to share adult content. However, this was a very small sample (5%) of actual data size. While a dominant feature was able to detect users who share negative content aggressively we wanted the model to be able to find user who share negative content in context. This will allow our pipeline to detect topics which co-occur with negative more frequently. E.g. health hazard like zika, ebola, or a latest controversy, scam, attack etc.

B. *Running clustering algorithm to find all users in negative cluster*

As shown in Figure 6, we tried hard clustering (K-Mean) and soft clustering (EM) on this feature data, and as expected, the EM worked better with high dimensional data. We will be presenting now an in-depth analysis of experiments. The pie chart of overall cluster distribution to only top 200 negative user sorted by negative topic feature value was very interesting.

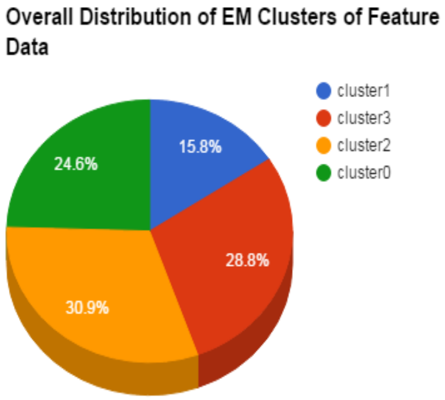


Figure 12 was very encouraging as EM was able to cluster all the top 200 users with 95.5% accuracy in single cluster; cluster1. This brought us to our next question:

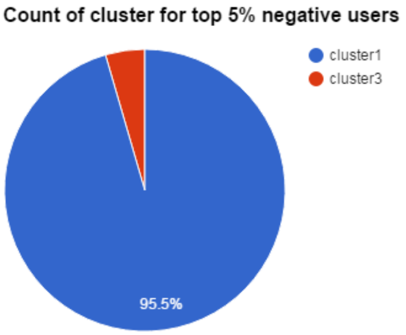


Figure 12: Overall distribution of cluster across 6000 user dataset against top 5% negative users.

While EM clustering algorithm was very good at detecting top 200 negative content distributor, how generic the algorithm was when it comes to all user who had their share in negative content distribution? That brought us to our next important question, how to label a user as negative user. For binary labeling a user as either N or NN we followed Algorithm 3 and for labeling a user based on topic contribution we followed Algorithm 2.

C. *Analysis after labeling users on binary classification*

Figure 7 suggests that (small arrow mark in image) standard deviation of F_9 (feature-9) is 0.06. Taking this as variable std-10, and feature vectors of all 6000 users as an input to Algorithm 3, we classified the instances in to binary classifiers, N/ NN. Also we did the analysis of how many of these instances classified as N, fell in to one cluster. Figure 13 suggests that, in 95.7 % cases, instances which were labeled as negative based on Algorithm 3, were part of single cluster, cluster 1. That also implies that cluster 1 had all (95% of them) the negative users at one place.

D. *Analysis after labeling users based on topic contribution*

We took the same feature vector as input and classified

Them as per topic distribution using Algorithm2 in to class 0-9 based on their dominant feature. Then we did the analysis of how many of these instances classified as "9" (9th topic being the adult content topic), fell in to one cluster. It was 100%! As shown in Figure 14.

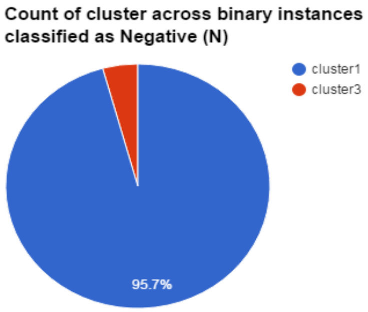


Figure 13 Cluster distribution on negative classified instances

This concludes out section that how we detected users with negative (aggressive or in general) content.

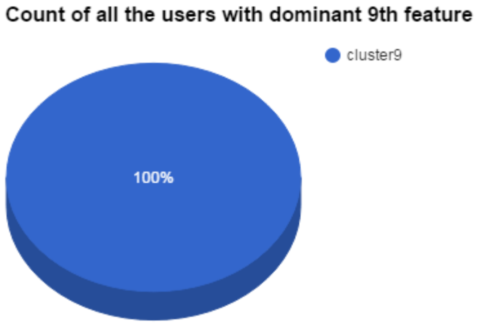


Figure 14: User's cluster distribution whose 9th feature was most dominant one

E. *Analysis of topic coemerging with negative topics.*

One of the motivation to analyze twitter data was also to find negative topics which were co emerging with the negative ones. Hence we started looking at feature data closely and fed them as input to t-SNE algorithm.

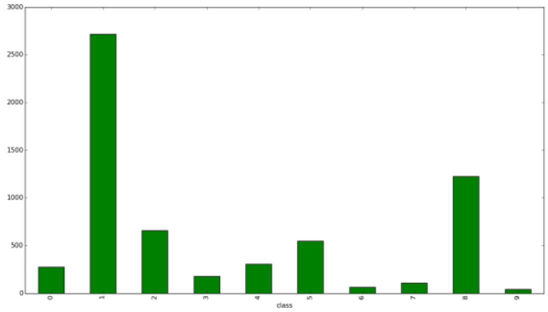


Figure 15 Data distribution across 10 features

As we can see in Figure 15 that feature 9 which is Dominant negative feature, is very less in number. However, there are lot of user who are using words appearing in topic 1 and 8. Hence we wanted to look at topic co-emerging with negative one. For that we fed the same feature data to t-SNE. The plot is as shown in Figure 16.

Figure 16 is mine of informtion. It depicts how topic like Topic5: Trump , Gay, Police, Kill and Topic4: pussy, amateur, milf, appear very close to Adult topics or negative topics, in comparison with positive words like good, love, thank, please appearing in topic 2. Now this property can be used in multiple ways. If rather than taking the regular stream, If we filter words by keyword like "cancer", a t-SNE on that will give us topic which are codiscussed with cancer and so on.

A system like this can be really useful to detect health hazards like ebola or zika. As the system can detect on its own that zika word is appearing a lot in negative context, as it detects Trump being used in negative context on social media more than positive.

F. *Topic weight and their correlation with t-SNE graph*

We ran all the topics file through Algorithm 5, and the result is depicted in Figure 17.

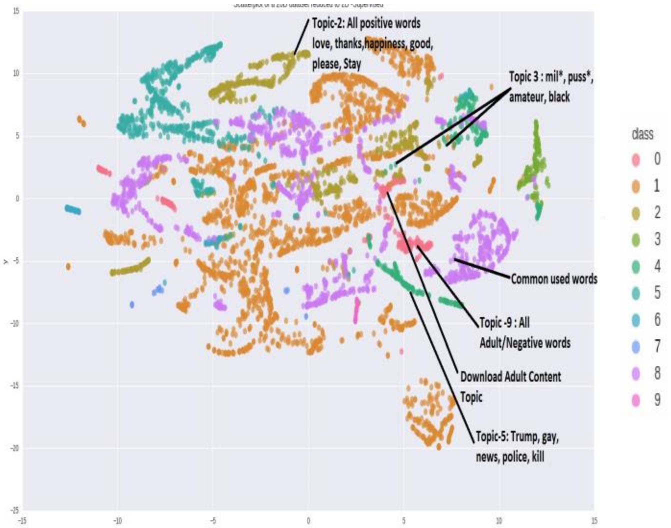


Figure 16 Distribution of 10-Dimensional data projected to 2-Dimension. Multiple topics are marked on image for showing distance concept.

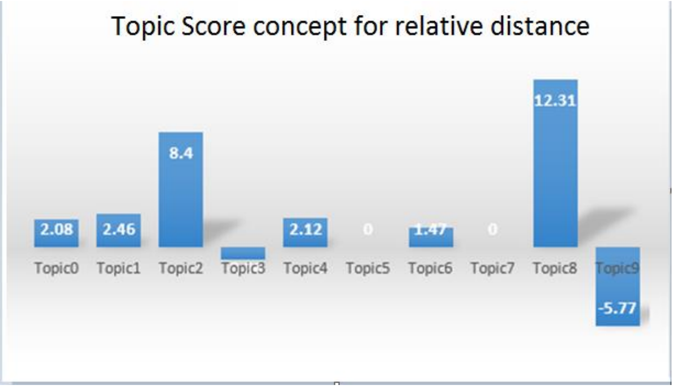


Figure 17 Topic score based on sentiment score and word probability. We can Topic 9 is most negative and among the topic where weight appears, Topic-3 has least. Even our t-SNE suggests the same.

This was a quick validation of our t-SNE approach a new concept of "Topic Score".

DISCUSSION

Our approach follows only text based analysis. The approach can be further improved by adding image/ video analysis. That will make us more accurate on a larger data set. We could also use some type of ontology to derive the topic context rather than manually looking at it for verification. Our Sentiment analysis approach is very naïve, hence we had to do third level of verification for accuracy, manually. Using a sophisticated sentiment analysis like Stanford NLP [25] can help us get rid of random manual validation.

CONCLUSION

In this paper, we propose an approach that exploits the LDA topic modeling to find users who share negative content on social media. What makes our approach impressive is accuracy and no bootstrapping. It's a pure unsupervised approach to learn about negative topics appearing on social media. Also as the approach gives a quantitative value of negativity for each users. This enables us practically to set a threshold values on which certain classifier criterion like high, medium, low contributor can be later assigned. For top 5% user we achieve detecting users who share adult content with an accuracy of 95.5%. This approach being generic in nature also allows us to find co-emerging negative topics. Like in this case framework was able to detect Trump being used in negative context more than positive on social media. We have also been able to coin a concept of topic score, which gives us a broader idea on what users to monitor. Even though paper runs through with number of topic being 10 as an example, we experimented with topics 20, 30, 40 and 50. In addition, our approach provides an alternative to designing large scale hand coded frameworks to detect changing events on social media hence replaces any supervised approach to achieve the same.

References

[1] Proportion of social media users to total internet users [online] http://wearesocial.com/sg/special-reports/digital-social-mobile-2015

[2] Report: Twitter hits half a billion tweets a day [Online]. Available: http://www.cnet.com/news/report-twitter-hits-a-billion-tweetsa-day

[3] Twitter now has more than 200 million monthly active users [Online]. Available: http://www.cnet.com/news/report-twitterhits-half-a-billion-tweets-a-day

[4] Hanqiang Cheng, Xinyu Xing, Xue Liu and Qin Lv, "ISC: An Iterative Social Based Classifier for Adult Account Detection on Twitter" IEEE transaction on knowledge and data engineering, April 2015

[5] Guang Xiang, Bin Fan, Ling Wang, Jason I. Hong, Carolyn P.Rose "Detecting Offensive Tweets via Topical Feature Discovery Over a Large Scale Twitter Corpus" Nov 2012

[6] Kyumin Lee, James Caverlee and Steve Webb, "Uncovering Social Spammers: Social Honeypots + Machine Learning" July 2010

[7] Gianluca Stringhini, Christopher Kruegel and Giovanni Vigna, Detecting Spammers on Social Networks Dec 2010

[8] Munmun De Choudhury, Scott Counts, Eric Horvitz, and Michael Gamon, "Predicting Depression via Social Media" 201

[9] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritte and Veselin Stoyanov, "SemEval-2015 Sentiment analysis in twitter" , 2015

[10] Kimberly McManus, Emily K. Mallory, Rachel L. Goldfeder, Winston A. Abstract Haynes, Jonathan D. Tatum, 2015, "Mining Twitter Data to Improve Detection of Schizophrenia "

[11] Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pages 36–44.

[12] AFINN data dictionary http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

[13] Opinion Mining, Sentiment Analysis, and Opinion Spam Detection, [online] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html ******

[14] Nagarajan Natarajan, Prithviraj Sen and Vineet Chaoji, "Community Detection in Content-Sharing Social Networks" 2013

[15] How and why the porn industry embraced twitter [Online]. Available: http://ibnlive.in.com/news/how-and-why-the-pornindustry-embraced-twitter/272639-11.html

[16] Porn stars use twitter to go mainstream [Online]. Available: http://www.cnn.com/2012/07/19/showbiz/porn-stars-twitter

[17] Kirtsy has porn, users blame twitter [Online]. Available: http:// thenextweb.com/2008/12/23/kirtsy-has-porn-users-blame-twitter

[18] An open letter to twitter: Stop the porn spam [Online]. Available: http://michellerafter.com/2009/07/08/an-open-letter-to-twitterstop-the - porn-spam

[19] Guess_language package for guessing the language of tweet text [online] https://bitbucket.org/spirit/guess_language

[20] Java language based data-structure to create unique text. [online] https://docs.oracle.com/javase/7/docs/api/java/util/HashSet.html

[21] Document term Matrix: [online] https://en.wikipedia.org/wiki/Document-term_matrix

[22] Gensim: Topic modeling for humans [online] https://radimrehurek.com/gensim/

[23] Visualization of SNPs with t-SNE, Alexander Platzer, Gregor Mendel Institute, Vienna, Austria

[24] Parametric nonlinear dimensionality reduction using kernel t-SNE, Gisbrecht A, Schulz A, Hammer B, Neurocomputing 2015

[25] Stanfor Deep Learning Sentiment analysis http://nlp.stanford.edu:8080/sentiment/rntnDemo.html

[26] Wang X, Garber MS and Brown DE, "Automatic Crime Prediction Using Events Extracted from Twitter Post s", 2012

[27] Prier KW, Smith MS, Giraud-Carrier C, Hanson_CL
Identifying Health-Related Topics on Twitter

[28] Michael Kearns, Yishay Mansour, Andrew Y. Ng , 2013, An Information-Theoretic Analysis of Hard and Soft Assignment Methods for Clustering

[29] Stop words [online] https://en.wikipedia.org/wiki/Stop_words

[30] Natural; Language Toolkit [online] http://www.nltk.org/

[31] The Porter Stemming Algorithm http://tartarus.org/martin/PorterStemmer/