ABSTRACT

Tumblr is one of the largest and most popular microblogging website on the Internet. Studies shows that due to high reachability among viewers, low publication barriers and social networking connectivity, microblogging websites are being misused as a platform to post hateful speech and recruiting new members by existing extremist groups. Manual identification of such posts and communities is overwhelmingly impractical due to large amount of posts and blogs being published every day. We formulate the problem of locating such extremist communities as a graph search problem. We propose a topical crawler based approach performing several tasks: searching for a blogger, computing its similarity against exemplary documents, filtering hate promoting bloggers, navigating through links to other bloggers and managing a queue of such bloggers for social network analysis. We conduct experiments on real world dataset and examine the effectiveness of 'like' and 'reblog' features as links between bloggers. Experimental results demonstrates that the proposed solution approach is effective with an F-score of 0.80. We apply social network analysis based techniques and identify influential and core bloggers in a community.

Categories and Subject Descriptors
H.5.4 [Hypertext/Hypermedia]: Navigation; K.4.2 [Social Issues]: Abuse and crime involving computers; D.2.8 [Metrics]: Performance measures; H.3.1 [Content Analysis and Indexing]: Linguistic processing

General Terms
Algorithm, Measurement, Performance, Legal Aspects, Graph Search

Keywords
Hate and Extremism Detection, Information Retrieval, Microblogging, Mining User Generated Content, Online Radicalization, Social Media Analytics, Social Network Analysis, Topical Crawler

1. INTRODUCTION

Tumblr is the second largest microblogging platform, has gained phenomenal momentum recently. It is widely used by fandoms: communities of users having similar interests in various TV shows and movies [12]. Therefore, it is especially popular among young generation users and provides them a platform to discuss daily events. They communicate by blogging and publishing GIF images as their reactions and emotions on several topics [2]. According to Tumblr statistics 2015[1] over 219 million blogs are registered on Tumblr and 420 million are the active users. 80 million posts are being published everyday, while the number of new blogs and subscriptions are 0.1 million and 45 thousands respectively.

Tumblr is also posed as a social networking website that facilitates users to easily connect to each other by following other users and blogs without having a mutual confirmation. Bloggers can also communicate via direct messages that can be sent privately or can be posted publicly using ‘ask box’. It facilitates bloggers to send these messages anonymously if they don’t want to reveal their Tumblr identity [10]. Similar to other social networking websites, Tumblr has very low publication barriers. A blogger can publish a new post and can re-blog an existing public post which is automatically broadcasted to it’s followers unless it is enabled as a private post [3]. The type of posts can be chosen among seven different categories including multi-media and other content: text, quote, link, photo, audio, video and URL. Unlike Twitter, Tumblr has no limit on the length of textual posts. Similar to hashtags in Twitter, there are separated tags associated with the blog content that make a post easier to be searchable on Tumblr [2]. Tumblr also allows users to update their other connected social networking profiles when something is posted.

The simplicity of navigation, high reachability across wide range of viewers, low publication barriers, social networking and anonymity has led users to misuse Tumblr in several ways. Previous studies shows that these features of Tumblr are exclusive factors to gain the attention of modern extremist groups [1][3]. This is because Tumblr provides every kind of multimedia posts which is a great medium to share your views with your target audiences. These groups form their own communities that share a common propaganda.

1 https://www.tumblr.com/about

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.
They post rude comments against a religion to express their hatred and spread extremist content. Social networking facilitates these groups to recruit more people to promote their beliefs and ideology among global audiences [5, 9]. Figure 1 illustrates a concrete example of various types of hate promoting posts and their associated tags on Tumblr. The number of notes shows the number of times that post has been liked and re-blogged by other blogs.

Online-radicalization and posting hateful speech is a crime against the humanity and mainstream morality; it has a major impact on society [6]. Presence of such extremist content on social media is a concern for law enforcement and intelligence agencies to stop such promotion in country as it poses the threat to the society [6]. It also degrades the reputation of the website and therefore is a concern for website moderators to identify and remove such communities. Due to the dynamic nature of website, automatic identification of extremist posts and bloggers is a technically challenging problem [14]. Tumblr is a large repository of text, pictures and other multimedia content which makes it impractical to search for every hate promoting post using keyword based flagging. The textual posts are user generated data that contain noisy content such as spelling, grammatical mistakes, presence of internet slangs and abbreviations. Presence of low quality content in contextual metadata poses technical challenges to text mining and linguistic analysis [10] [11]. The work presented in this paper is motivated by the need of investigating solutions to counter and combat the online extremism on Tumblr.

The research aim of the work presented in this paper is the following:

1. To investigate the application of topical crawling based algorithm for retrieving hate promoting bloggers on Tumblr. Our aim is to examine the effectiveness of random walk in social network graph graph traversal and measuring its performance.
2. To investigate the effectiveness of contextual metadata such as content of the body, tags and caption or title of a post for computing the similarity between nodes in graph traversal. To examine the effectiveness of reblogging and like on a post as the links between two bloggers.
3. To conduct experiments on large real world dataset and demonstrate the effectiveness of proposed approach in order to locate virtual and hidden communities of hate and extremism promoting bloggers and apply Social Network Analysis based techniques to locate central and influential users.

2. LITERATURE SURVEY

In this section, we discuss closely related work to the study presented in this paper. Based on our review of existing work, we observe that most of the researches for detecting online radicalization are performed on Twitter, YouTube and various discussion forums. We conduct a literature survey in the area of identifying hate promoting communities on social networking websites and short text classification of Tumblr microblog. O’Callaghan et. al. [11] describe an approach to identify extreme right communities on multiple social networking websites. They use Twitter as a possible gateway to locate these communities in a wider network and track dynamic communities. They perform a case study using two different datasets to investigate English and German language communities. They implement a heterogeneous network within a homogeneous network and use four different social networking platforms (Twitter accounts, Facebook profiles, YouTube channels and all other websites) as extreme right entities or peers and edges are the possible interactions among these accounts.

Mahmood S. [9] describes several mechanisms that can be useful in order to detect presence of terrorists on social networking websites by analyzing their activity feeds. They use Google search and monitor terror attack using keyword-based flagging mechanism. They monitor sentiments and opinions of users following several terrorism groups on online social networks and propose a counter-terrorism mechanism to identify those users who are more likely to commit a violent act of terror. They also discuss honeypots and counter-propaganda techniques that can be used to rehabilitate radicalized users back to normal users. The advantage of keyword based flagging approach is the large number of false alarms. David and Morcelli [5] present a keyword based search to detect several criminal organizations and gangs on Twitter & Facebook. They discuss a study of analyzing the presence of organized crime and how these gangs
use social media platforms to recruit new members, broadcast their messages and coordinate their illegal activities on web 2.0. They perform a qualitative analysis on 28 groups and compare their organized crime between 2010 and 2011 on Facebook.

Agarwal et. al. propose a one-class classification model to identify hate and extremism promoting tweets [13]. They conducted a case study on Jihad and identified several linguistic and stylistic features from free form text such as presence of war, religious, negative emotions and offensive terms. They conduct experiments on large real world dataset and demonstrate a correlation between hate promoting tweets and discriminatory features. They also perform a leave-p-out strategy to examine the influence of each feature on classification model.

In context to existing work, the study presented in this paper makes the following unique contributions:

1. We present an application of topical crawler based approach for locating extremism promoting bloggers on Tumblr. While there has been work done in the area of topical based crawling of social media platforms, to the best of our knowledge this paper is the first study on topical crawling for navigating connections between Tumblr bloggers.

2. We conduct experiments on large real world dataset to demonstrate the effectiveness of one class classifier and filtering hate promoting blog posts (text). We retrieve Tumblr blogger profiles and their links with other hate promoting bloggers and apply Social Network Analysis to locate strongly connected communities and core bloggers.

3. EXPERIMENTAL SETUP

In a graph traversal, a topical crawler returns relevant nodes to a specific topic. To define the relevance of a node it learns the characteristics and features of given topic and computes the extent of similarity against a bunch of exemplary documents. Figure 2 illustrates the general framework to obtain these documents. As shown in Figure 2 to collect these training examples, we perform an iterative search on Tumblr using keyword based flagging, where keyword is a search tag; for example, jihad, anti-islam and hate. We perform a case study on Jihad and by manual search on Tumblr posts we collect several relevant tags that are commonly used by extremist bloggers. We use these tags to initiate our process and collect all textual posts (avoiding picture, audio, video and URLs), tags (associated with resultant posts) and linked bloggers (post reblogged by and liked by) with no redundancy. We perform a manual inspection on resultant
posts and posts made by linked bloggers to filter relevant (hate promoting) and unknown results. We further extract more posts and linked bloggers from related tags and run this framework recursively to collect our exemplary documents. These training examples contain the body and caption of only positive class (hate and extremism promoting content) posts which is used to train the model. During the manual inspection of exemplary documents and resultant posts we observe many keywords that are frequently used by extremist bloggers. Figure 4 shows a word cloud of such terms.

We use these linked bloggers and posts to compute the threshold value for language modeling. We take a sample of 30 bloggers and compare their posts with the exemplary documents. For each blogger we get a relevance score. To compute the threshold value for similarity computation we take an average of these scores. Figure 5 illustrates the relevance score statistics of each blogger (Sorted in increasing order). We notice that 80% of the bloggers have relevance score below -2.7 and -1.5. We take average (turns out to be -2.58) of these scores to avoid the under-fitting and over-fitting of bloggers during classification.

4. RESEARCH METHODOLOGY

In this section, we present the general research framework and methodology of proposed approach for classifying extremist bloggers on Tumblr. Figure 3 illustrates the design and architecture of topical crawler to locate extremist communities. As shown in Figure 3, solution framework is an iterative multi-step process primarily consists of five phases: features (posts) extraction, data pre-processing, classification, frontier extraction and graph traversal. In phase 1, we initiate our process using a positive class (hate promoting) blogger \( U_i \) called as ‘seed’. We use Tumblr API\(^3\) to fetch the URLs of \( n \) number of textual posts and by using Jsoup Java library\(^4\) we extract the content and caption of these posts (used as contextual metadata). These posts can be either re-blogged from other users or originally posted by the user \( U_i \). These posts consist of multiple languages. Therefore, in phase 2, we perform data pre-processing and filter English and non-English posts using language detection library\(^5\). We perform data pre-processing on these posts and remove English stopwords. In phase 3, we build a statistical model from the exemplary documents collected separately by semi-automatic process (refer to Figure 2). To compute the relevance of each blogger, we use character level n-gram language modeling approach. We find the extent of similarity between metadata and exemplary documents using LingPipe API\(^6\). We implement a one class classifier and filter extremism promoting bloggers from unknown bloggers.

In phase 4, we extract the notes associated with the posts (collected in phase 1) of relevant bloggers. These notes contain the list of bloggers who liked and re-blogged a particular post. The number of notes represent the popularity of a post and indicate the similar interest between original poster and other bloggers in the list who may or may not be the direct followers of each other. We use notes to extract frontier nodes of a blogger because of two reasons: 1) due to the privacy policies Tumblr API does not allow developers to extract followers and following blogs of Tumblr users. 2) Tumblr facilitates bloggers to track any number of tags so that whenever there is a new post published publicly on Tumblr containing any of these tags, it automatically appears in a menu on user’s dashboard. They can spread that post among their followers by re-blogging it. Tracked tags allow bloggers to form a virtual community without following each other. For each frontier extracted in phase 4, we compute the relevance score against exemplary documents and discard unknown bloggers. In phase 5, we manage a queue of relevant bloggers and perform directed graph traversal using random walk algorithm. To expand our graph we select the next blogger in uniform distribution and extract it’s frontiers. We execute our focused crawler for each frontier without revisiting a blogger. This traversal results in a connected graph, where nodes represents a blogger (hate promoting) and edges represent the links (re-blog and like) between two bloggers. We perform social network analysis on the resultant graph and locate extreme right communities of hate promoting bloggers.

5. SOLUTION IMPLEMENTATION

A topical crawler starts from a seed node, traverses in a graph navigating through some links and returns all relevant nodes to a given topic. In proposed solution approach we divide our problem into three sub-problems. First we classify the given seed node \( S \) as hate promoting or unknown according to the published post (originally posted by blogger or re-blogged from other Tumblr users). Second, if the node is relevant then we extend this node into it’s frontiers and it further leads us to more hate promoting bloggers. In third sub-problem, we perform topical crawling on Tumblr

---

\(^3\)https://www.tumblr.com/docs/en/api/v2
\(^4\)http://jsoup.org/apidocs/
\(^5\)https://code.google.com/p/language-detection/
\(^6\)http://alias-i.com/lingpipe/index.html
network and use random walk algorithm to traverse along the graph.

5.1 Retrieval of Published Posts

Algorithm 1 describes the method to search Tumblr posts using keyword based flagging and extraction of posts published by a given blogger. The work presented in this paper focus on mining textual metadata on Tumblr therefore we set few parameters and extract only text based posts for further analysis. For each blogger we set the limit of 100 posts published recently. Function SetParameters() (steps 13 and 14) filters the search results and displays only the textual posts. Function TaggedPost() with given parameters search for text posts that exclusively contain given tag name. BloggerPost() fetches the textual posts published by given blogger name. Both the functions make a Tumblr API request to fetch these data. Function ExtractPost() filters the response and extract body content & caption of each post. Tumblr API allows us to only extract the summary of large posts. Therefore we use HTML parsing for extracting the whole message in blog post. In steps 4 to 7, we generate the URL from post summary and id to fetch the remaining post details. ID is a unique identifier of Tumblr user. The link between two bloggers indicates the similarity between them so that number of frontiers vary for every post published by a blogger. For each user, we extract 25 bloggers for each relation i.e. users who have liked and re-blogged that post recently. If a blogger B₂ re-blogged the post recently. To extract the linked bloggers of a Tumblr user we first need to extract the posts made by U. We can extract notes information only when notes and re-blog information parameters are set to be true (refer to steps 15 and 16 of Algorithm 1). As described in Algorithm 2 in step 4, we extract notes for each textual post (hate promoting) made by User U. In steps 5 to 8, we extract the number of hit counts on their recent 100 posts. These number of notes varied from 0 to 25K therefore we perform smoothing on datapoints and plot median of these values. Figure 6 shows the statistics of number of notes collected on 100 posts of each blogger extracted during topical crawler. Figure 6 reveals that overall number of hit counts (number of reblog and like) for extremism promoting users is very high. These hit counts reveals the popularity of extremist content and the number of viewers connected to such bloggers.

5.2 Retrieval of External Links to Bloggers

Algorithm 2 describes the steps to extract frontiers of a given node. Due to the privacy policies, Tumblr API does not allow developers to extract subscriptions and followers of a Tumblr user. The link between two bloggers indicates the similarity between them so that number of frontiers vary for every post published by a blogger. For each user, we extract 25 bloggers for each relation i.e. users who have liked and re-blogged that post recently. If a user B₂ re-blogged a post as well as likes a post published by another blogger B₁ then in the graph G, we create an edge with both labels i.e. (B₁, B₂, "like, re-blog"). To avoid the redundancy we extract one more frontier who have either liked or re-blogged the post recently. In step 8, F represents the list of frontiers and relation of U with each frontier. We maintain a list of all processed bloggers and the number of hit counts on their recent 100 posts. These number of notes varied from 0 to 25K therefore we perform smoothing on datapoints and plot median of these values. Figure 6 reveals that overall number of hit counts (number of reblog and like) for extremism promoting users is very high. These hit counts reveals the popularity of extremist content and the number of viewers connected to such bloggers.

5.3 Topical Crawler Using Random Walk

Algorithm 3 describes the proposed crawler for locating a group of hidden extremist bloggers on Tumblr. The goal of this algorithm is to compare each blogger against training examples and then connecting all positive class (hate

Algorithm 1: Extracting Textual Posts on Tumblr

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data:</strong> User U, Consumer Key C₁, Consumer Secret C₂, Search Tag tag name</td>
<td></td>
</tr>
<tr>
<td><strong>Result:</strong> Text based posts made by User U or associated with tag tag name</td>
<td></td>
</tr>
<tr>
<td><strong>Algorithm</strong> ExtractPost()</td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>SetParameters()</td>
</tr>
<tr>
<td>2.</td>
<td>select a method to extract posts TaggedPost() OR BloggerPost()</td>
</tr>
<tr>
<td>3.</td>
<td>Generate URL of post to fetch post content and caption</td>
</tr>
<tr>
<td>4.</td>
<td>for all postP ∈ Posts do</td>
</tr>
<tr>
<td>5.</td>
<td>Slug = P.getSlug()</td>
</tr>
<tr>
<td>6.</td>
<td>id = P.getId()</td>
</tr>
<tr>
<td>7.</td>
<td>URL = &quot;<a href="http://blog_name.tumblr.com/post/id">http://blog_name.tumblr.com/post/id</a> slug&quot;</td>
</tr>
<tr>
<td>9.</td>
<td>post_content = Document.getDescription()</td>
</tr>
<tr>
<td>10.</td>
<td>post_caption = Document.getTitle()</td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td><strong>Algorithm</strong> SetParameters()</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>Authenticate the client via API Keys C₁ and C₂</td>
</tr>
<tr>
<td>12.</td>
<td>Set the parameters</td>
</tr>
<tr>
<td>13.</td>
<td>params.put(&quot;type&quot;, &quot;text&quot;)</td>
</tr>
<tr>
<td>14.</td>
<td>params.put(&quot;reblog info&quot;, true)</td>
</tr>
<tr>
<td>15.</td>
<td>params.put(&quot;notes info&quot;, true)</td>
</tr>
<tr>
<td><strong>Algorithm</strong> TaggedPost(tag name)</td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td>Posts = client.tagged(tag name, params)</td>
</tr>
<tr>
<td>17.</td>
<td>end</td>
</tr>
</tbody>
</table>

Algorithm 2: Extracting Frontiers of a Given Blog

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data:</strong> Blogger U</td>
<td></td>
</tr>
<tr>
<td><strong>Result:</strong> Frontiers F &lt; name, type &gt; of U</td>
<td></td>
</tr>
<tr>
<td><strong>Algorithm</strong> ExtractFrontiers(U)</td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>SetParameters()</td>
</tr>
<tr>
<td>2.</td>
<td>Posts = BloggerPost(U)</td>
</tr>
<tr>
<td>3.</td>
<td>for all postP ∈ Posts do</td>
</tr>
<tr>
<td>4.</td>
<td>Notes = P.getNotes()</td>
</tr>
<tr>
<td>5.</td>
<td>for all NoteN ∈ Notes do</td>
</tr>
<tr>
<td>6.</td>
<td>Linked_Blog_Name = N.getBlogName()</td>
</tr>
<tr>
<td>7.</td>
<td>Note_Type = N.getType()</td>
</tr>
<tr>
<td>8.</td>
<td>Liked or Reblog</td>
</tr>
<tr>
<td>9.</td>
<td>Frontiers F.add(Linked_Blog_Name, Note_Type)</td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6: Illustrating the Number of Notes For Each Blogger Traversed in Topical Crawler
**Algorithm 3:** Graph Traversal Using Random Walk Algorithm

Data: \( S, th, N_s, S_g, W_g, D_s \)

Result: Directed Graph \( G \)

1. **SetParameters()**

2. \( U_i = S, F.add(S) \)

3. **TopicalCrawler(S)**

   while (\( |graphsize| < S_g \) OR \( F.size > 0 \)) do

   4. Posts = ExtractPost(U_i)

   5. \( \text{Relevance\_Score} = \text{LanguageModeling}(D_s, Posts, N_s) \)

   6. if (\( \text{score} > th \)) then

      7. \( \text{Linked\_Users} = \text{ExtractFrontiers}(U_i) \)

      8. \( \text{Processed\_Nodes} \) \( PN.add(U_i) \)

      9. for all \( LU \in \text{Linked\_Users} \) do

         10. if (\( (F.contains(LU)) \) AND (\( PN.contains(LU) \))) then

            11. \( F.add(LU) \)

         12. else

            13. Discard the node \( LU \)

      end

   else

      14. Discard the node \( U_i \)

   end

15. Compute the Markov Chain over graph \( G \)

16. New\_Blogger = node with maximum probability in Markov chain array

17. F.remove(New\_Blogger)

18. \( \text{TopicalCrawler}(\text{New\_Blogger}) \)

end

promoting or relevant) bloggers. Algorithm 3 takes several inputs: seed blogger (positive class user) \( S \), size of the graph \( S_g \) i.e. maximum number of nodes in a graph, width of the graph \( W_g \) i.e. the maximum number of frontiers or adjacency nodes for each blogger, a set of exemplary documents \( D_s \), threshold \( th \) and n-gram value \( N_s \) for relevance computation. We create a list of 30 positive class bloggers extracted during experimental setup (refer to section 5) and compute their relevance score against the exemplary documents. We take an average of these scores and compute the threshold value for language modeling. We use n-gram language modeling (\( N_g = 3 \)) to build our statistical model. Algorithm 3 is a recursive process that results into a cyclic directed graph. We run this algorithm until we get a graph of size \( S \) (1000 bloggers) or there is no node left in the queue for further extension. We perform a self-avoiding random walk that means we make sure a node is never being re-visited. If a node re-appears in the frontiers list then there are two possibilities: 1) the frontier has already been processed (extended or discarded based upon the relevance score- Steps 4 to 7). If it exists in the processed nodes list then we create a directed edge between the node and it’s parent and avoid further extension. 2) If the re-appearing node is in frontiers list and is not yet processed, we created a directed edge in the graph and continue the traversal.

The topical crawler is a recursive process that adds and removes nodes after each iteration. The resultant graph is dynamic and not irreducible that means given a graph \( G(V, E) \), if there is a directed edge between two nodes \( u \) and \( v \), it is not necessary that there exists a directed path from \( v \) to \( u \). Consider that object (topical crawler) processed node \( i \) at time \( t - 1 \). In the next iteration object moves to an adjacency node of \( i \). The probability that object moves to node \( j \) at time \( t \) is \( \frac{1}{d_j^k} \) when there exists a direct edge from \( i \) to \( j \). \( M_{ij} = \frac{1}{d_j^k} \) denotes the probability to reach from \( i \) to \( j \) in one step where \( d_j^k \) is the out-degree of node \( i \). Therefore we can define:

\[
M_{ij} = \begin{cases} 
\frac{1}{d_j^k}, & \text{if } (i, j) \text{ is an edge in digraph } G \\
0, & \text{otherwise}
\end{cases}
\]

Therefore for each vertex \( i \), the sum of the probability to traverse an adjacency node of \( i \) is 1.

\[
\forall i \sum_j M_{ij} = 1
\]

Where, \( A(i) \) denotes the list of adjacency nodes \( i \). In random walk on graph \( G \) topical crawler traverse along the nodes according to the probability of \( M_{ij} \). Graph \( G \) is a dynamic social networking graph, therefore we compute a Markov chain \( M \) after each iteration and compute the probability matrix over graph \( G \). Markov chain is a random process where the probability distribution of node \( j \) depends on the current state of matrix. The probability matrix \( M^K \) gives us a picture of graph \( G \) after \( k \) iterations of topical crawler. Using this matrix, we compute the probability distribution \( P \) that object moves to a particular vertex. \( P^K \) is the probability distribution of a node \( j \) after \( k \) iteration then probability of \( i \) to be traversed in \( k + 1^{th} \) iteration is the following:

\[
P^{k+1} = P^k \cdot M \quad \text{where} \quad P^k = P^0 \cdot M^k
\]

Where \( P^0 \) is the initial distribution fixed for the seed node.

### 6. EXPERIMENTAL RESULTS

#### 6.1 Topical Crawler Results

We execute our topical crawler for a given seed blogger and traverse through Tumblr network using random walk algorithm. For every new blogger, we compute its relevance and classify it as hate promoting or unknown using one class classifier. To examine the effectiveness of our classifier, we compute its accuracy using standard information retrieval techniques. In one execution of our topical crawler, we were able to collect 600 bloggers. We hired 30 graduate students as volunteers from different department to label these bloggers as hate promoting or unknown according to their published posts and given guidelines for annotation. To avoid the biasness and to collect correct annotated results we perform a horizontal and vertical partition on nodes and arrange these 600 bloggers into a 2D matrix where rows are the numbers of annotators grouped in 10 sets, 3 members each. Columns of the matrix are the number of bloggers assigned to each member for annotation i.e. 60. We use majority voting approach for final annotation, the class of a blogger is the one which is voted by at least two annotators. Based upon the validation results we evaluate the accuracy of our model. Table [1(a)] shows the confusion matrix for one class classification. Table [1(a)] reveals that our model predicts 382(290+92) bloggers as hate promoting and 218(173+45) bloggers as unknown. Table [1(a)] shows that there is a missclassification of 13% and 34% in predicting hate promoting and unknown bloggers. Table [1(b)] shows the accuracy results of our classifier. Results shows that the precision, recall and f-score are reasonably high and we are
Table 1: Confusion Matrix and Accuracy Results for One Class Classifier

(a) Confusion Matrix

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
<th>Positive</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>290</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>Unknown</td>
<td>92</td>
<td>173</td>
<td></td>
</tr>
</tbody>
</table>

(b) Accuracy Results

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.75</td>
<td>0.86</td>
</tr>
<tr>
<td>F-Score</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.77</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Illustrating The Network Level Measurements for Topical Crawler. Dia= Network Diameter, Mod= Modularity, ACC= Average Clustering Coefficient, IBC= In- Betweenness Centrality, CC= In- Closeness Centrality, #SCC= Number of Strongly Connected Components.

<table>
<thead>
<tr>
<th>Graph</th>
<th>#Nodes</th>
<th>#Edges</th>
<th>Dia</th>
<th>#SCC</th>
<th>#ACC</th>
<th>#Mod</th>
<th>IBC</th>
<th>ICC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC</td>
<td>382</td>
<td>275</td>
<td>4</td>
<td>137</td>
<td>0.026</td>
<td>12.00</td>
<td>11.36</td>
<td>0.20</td>
</tr>
<tr>
<td>LB</td>
<td>27</td>
<td>60</td>
<td>1</td>
<td>21</td>
<td>0.0231</td>
<td>1.307</td>
<td>0</td>
<td>0.38</td>
</tr>
<tr>
<td>RB</td>
<td>355</td>
<td>215</td>
<td>6</td>
<td>185</td>
<td>0.021</td>
<td>7.01</td>
<td>6.284</td>
<td>0.40</td>
</tr>
</tbody>
</table>

Figure 7: Cluster Representation of Social Network Graphs- Topical Crawler using Random Walk (a), 'Posts Liked by' (b) and 'Posts Re-blogged By' (c)

able to predict hate promoting bloggers with an accuracy of 77%.

6.2 Social Network Analysis

We perform social network analysis on topical crawler’s network resulted into a directed graph $G(V, E)$, where $V$ represent a set of Tumblr bloggers accounts and $E$ represent a directed edge between two bloggers. We define this edge as a relation having two labels ‘posts liked by’ and ‘post re-blogged by’. To examine the effectiveness of these relations we generate two independent networks exclusively for ‘like’ and ‘re-blog’ links between bloggers. Figure[7] illustrates the representations of these networks. In each graph, size of the node is directionally proportional to its out-degree. A node with maximum number of adjacency vertices is biggest in size. Colors in the graph represents the clusters of nodes having similar properties. Here, we define the similarity measure as the ratio of out-degree and in-degree.

We also perform several network level measurements on these graphs. Table[2] reveals that re-blogging is a good indicator of connection between two bloggers. Here, we observe that the graphs generated for topical crawler and re-blogging link have same pattern in network measurements. Both graphs are dense (also evident from the Figure[7]) and have higher modularity in comparison to the graph created for ‘liked’ link. Table[2] and Figure[7] also reveal that by navigating through re-blogging links we can locate large number of connected components in a extreme right community. While following ‘like’ as a link we are able to detect small number of connected blogs. Though as illustrated in Figure[7(b)] we can not completely avoid this feature as a set of blogs extracted using this link are irreducible. Table[2] also shows that the graph created for ‘like’ relation has slightly larger value for average clustering co-efficient. This is because the number of nodes in the graph is very less and a major set of these nodes is strongly connected. Higher value of In-between centrality shows the presence of bloggers who are being watched by a large number of users. As the Figure[7(c)] shows there are many users which are not directly connected to each other (shown in red color) but has a huge network of common bloggers. These disjoint bloggers are two or three hop away and are connected via other bloggers (having second largest number of adjacency nodes). These bloggers are connected with maximum number of other bloggers present in the graph and has a wide spread network in extreme right communities. These nodes have the maximum
closeness centrality and play central role in the community. Nodes represented as black dots have minimum number of out-degree. They don’t have a directed path to the central users or original source of extremist posts. Based upon our study we find these bloggers to be the target audiences who share these posts in their own network. These users are very crucial for such communities though they don’t play a major role in the network.

7. CONCLUSIONS

In this paper, we perform a case study on Jihadist groups and locate their existing extreme right communities on Tumblr. We conduct experiments on real world dataset and use topical crawler based approach to collect textual data (published posts) from Tumblr users. We perform one class classification and identify hate promoting bloggers according to the content present in their posts. We use random walk algorithm for graph traversal and extract exclusive links to these bloggers. We conclude that by performing social networking analysis on a graph (vertices are the Tumblr bloggers and edges are the links among these bloggers: re-blog and like) we are able to uncover hidden virtual communities of extremist bloggers with an accuracy of 77%. We locate users who are central and influential among all and play major role in the communities. We perform independent social network analysis on like and re-blog links among bloggers and conclude that re-blogging is a discriminatory feature that plays crucial role in the network.

We perform a manual inspection on Tumblr and perform a characterization on several hate promoting posts. Our study reveals that these posts are very much popular among extremist bloggers and get large number of hits. These posts are published targeting some specific audiences. Keywords present in the blog content, tags associated with post and comments by other bloggers are clear evidence of hate promotion among their viewers.

8. REFERENCES