TEA: Episode Analytics on Short Messages

Prapula G
Center for Data Engineering
IIIT Hyderabad
Andhra Pradesh, India
prapula.g@research.iiit.ac.in

Soujanya Lanka
Center for Data Engineering
IIIT Hyderabad
Andhra Pradesh, India
soujanya@iiit.ac.in

Kamalakar Karlapalem
Center for Data Engineering
IIIT Hyderabad
Andhra Pradesh, India
kamal@iiit.ac.in

ABSTRACT

Twitter is a widely used micro-blogging service, which in recent times, has become a reliable source of happening news around the world [11]. Breaking news are covered in twitter; the magnitude and volumes of tweets reflecting on the nature and intensity of the news. During events, many tweets are posted either expressing sentiments about the event or just about the occurrence of the event. Events related to an entity that have attracted a large number of tweets can be considered significant in the entity's twitter lifetime. Entity could represent a person, movie, community, electronic gadgets, software products and like wise. In this work, we attempt to automatically detect significant events related to an entity. An episode, is an event of importance; identified by processing the volumes of tweets/posts in a short time.

The key features implemented in Tweet Episode Analytics (TEA) system are: (i) detecting episodes among the streaming tweets related to a given entity over a period of time (from the entity's birth i.e., mention in the tweet world till date), (ii) providing visual analytics (like sentiment scoring and frequency of tweets over time) of each episode through graphical interpretation.

Categories and Subject Descriptors

H.4 [**Web IR and Social Media Search**]: Social Network Analysis(Micro-Blogging Analysis)

General Terms

Entity, Trend, Events, Sentiment, Analysis, Detection

Keywords

Tweets, *Episode*, Text Analytics

1. INTRODUCTION

Tweets are a source of valuable information that have the potential of providing an overview of how the world is thinking about various events/persons over a period of time. The events are usually related to nouns like persons, movies and objects in real world; these nouns are referred to as entities. Each entity will have a series (one or more) of events which are significant in its lifetime. People tweet about events that are of importance to them[16][13]. People seek latest up-to-date information by searching through tweets live stream. So, an event or a search phrase obtains a high frequency of tweets, mostly due to its significance (like a trending topic). Hence, the overall social interest received for an event related to an entity is reflected by the number of tweets that mention the event. This streaming information about various events should be identified, analyzed and visualized in order to make them suitable for humans to understand and interpret the causes and the consequences. Such a visual representation is also useful in displaying search results. AspecTiles[10] address the problem of search result diversification. In our work, *given an entity we address the event diversification related to an entity.* For instance, if a search on 'Roger Federer' is performed during the Wimbledon season, there could be various events related to Federer that would have been tweeted on different days of the season. Identifying significant events and displaying sets of tweets (by grouping tweets related to a particular event) with graphs gives user a chance to glance through events and explore in detail on an event he/she is interested in.

With large number of twitter users getting interested in a particular event leads to a deluge of tweets and also the queries on those tweets. Mining significant events will be useful in summarizing the deluge of tweets. Hence, an analysis system is needed, that (i) identifies important events related to an entity, (ii) analyzes the temporal sentiment patterns of tweets during the period of increased interest and provides visuals depicting the same. A large scale processing is done to accomplish all of this and the results of each of the above is presented in Section 5.

The importance of an event can be computed by the frequency of tweets and re-tweet counts related to the event as done in [14]. A popular entity (like a movie star, movie, musician and the likes) receives some amount of attention on a regular basis in twitter. The amount of attention received need not to be constant over a daily basis. The attention received (i.e., the number of tweets talking about the entity) varies over a period of time due to various events related to the entity. When there is a spike in the attention received, the event associated could be a significant one.

For instance, let us consider 'Lady Gaga' as an entity. There could be many tweets that mention Lady Gaga as part of routine events like '@user432 Listening to Lady Gaga',

Published as part of the #Microposts2014 Workshop proceedings, available online as CEUR Vol-1141 (http://ceur-ws.org/Vol-1141)

#Microposts2014, April 7th, 2014, Seoul, Korea.

'just read article on Lady Gaga', 'Lady Gaga in Japan' and 'Lady Gaga's Born this way - releasing in 2012'. Among these, significant events for Lady Gaga could be 'Born this way' album's release and her 'tour to Japan'. A significant event due to increased volumes of tweets related to an entity is considered as an episode.

The sentiments expressed by twitter users about episodes change over time. For example, there could be a very positive anticipation for a particular movie about to be released, but it might not have been well received (paving way for negative sentiments expressed post-release). Analyzing and visualizing the accumulated sentiments about episodes over time could be useful for market research analysis of an entity (movies, electronic gadgets, albums etc).

In this paper, we introduce the concept of an episode for a time-line of an entity and develop a tweet episode analytics system (referred to as TEA) which when given a phrase of words that represent an entity as input can: (a) identify episodes, (b) analyze episodes, life-spans, (c) display the cumulative sentiments expressed over a period of time.

In section 2, we present related work. In section 3, an Overview of TEA is presented which is followed by Tweet Episode Analytics (Section 4). Section 5 presents Results of TEA with Section 6 presenting some conclusions.

2. RELATED WORK

There has been a considerable amount of work done on extracting trending topics from twitter. The idea of an Episode that has been proposed in this paper is different from the past studies on trending topics. There has been a study on how and why the topics become trending in one of the papers [6]. As a part of their study, [6] have tried to explain the growth of trending topics. They have concluded that most topics do not trend for long on Twitter. This conclusion from their study strengthens our idea of Episodes which we have defined as a significant event that may occur in the time line of an entity and the event will be significant only for a short period of time.

In [7], Becker *et al* identified real-world events and their associated twitter messages that are published. Online clustering and filtering framework is used to address this event identification problem.We have introduced the concept of an episode and have presented an algorithm to identify an episode by considering accumulated significance of the tweets.

In [14], Nichols *et al* extracted sporting events and summarized the tweets in that events. They are confined to tweets related to sports and concentrated more on summarizing than extracting events. Our frame work and algorithm work for a search query (to represent the entity) and detect possible episodes in its life time.

In [15], Sakaki *et al* believe that when a real event like natural disasters that influence people from either one region or some parts of the world occur, the twitter users (social sensors) will tweet about the event immediately. This paper aims to recognise events at real time whereas we detect episodes that have already occurred and have lots of importance in the entity's life time. Our paper presents historical coverage of an entity as a sequence of episodes. Moreover, this paper targets events like social events (e.g., large parties), sports events, accidents and political campaigns and natural events like storms, heavy rainfall, tornadoes, typhoons which influence people's daily life whereas our work is not specific to any event of an entity and is more generic.

In [9], Gruhl *et al* studied the propogation of information in environments like personal publishing using a large collection of web logs. They have characterized the topics into long running "chatter" topics consisting of recursive "spikes" topics. According to their theory, if there are spikes recursively for a topic over a long period of time, it may be of interest. Topics are detected and then classified if its chatter or spike and studied the propagation. Our work concentrates on detecting events related to an entity based on a similar notion that spikes are the places where significant events have occurred in an entity's life time.

3. OVERVIEW OF TEA

In this section, we introduce the concept of an episode. We also present the architecture of "Tweet Episode Analytics" system as a part of this section.

3.1 What is an *Episode*?

Episode can be defined as a significant event in the time line of an entity (individual person, community, group etc) that has occurred due to a sudden increase of tweet volumes of the entity from its regular volumes.

Among all the events that an object/entity is involved in, the events that received more attention in a particular period of time, are referred as episodes. All episodes are events but not all events can be episodes. Episodes are significant events with respect to an entity, but events are more general not specifically related to entities. Episodes are always for an entity. TEA algorithm identifies prominent episodes of an entity that has occurred over its time line, considering an entity has a long lifespan. An episode is different from the traditional concept of "a trending topic" [12] or "topics extracted from topic clustering" [8]. An entity is said to have an episode if there is a sudden spike in an activity and that is captured as an event in the time line of the entity because of which there is a huge activity related to the entity. For each such event, there is evidence like an article or information that shows the true importance of the event. If no such article or information exists, then it may not be an episode.

Similar to 'Lady Gaga' example mentioned in Section 1, we noticed a similar episode being detected in our tweet data set related to 'Justin'(entity). A phrase formed by 'Justin' and 'Boyfriend' put together is an episode whereas 'Justin' is not. After the release of Justin Bieber's new song 'Boyfriend', there was a sudden outburst of tweets about this song. Even though the number of tweets about 'Justin' are large implying that it is a trending topic, it is not an episode because the reason for more social activity about 'Justin' is not due to a single significant event.

3.2 System Architecture

The whole tweet episode analytics system can be divided into different modules. Tweet collection and tweet processing are offline modules (module in which processing is done beforehand) where as, episode detection, sentiment analyzing are online modules (module in which the processing starts after receiving the query as input to the system). The flowchart of system architecture to "Detect Episodes of an entity from Twitter data using Episode Detection Algorithm" is given in Figure 1. Below is a brief explanation for each of the modules.

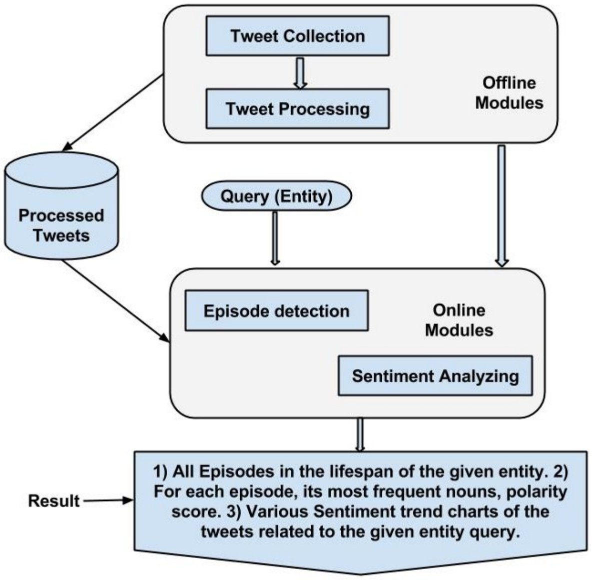

Tweet Collection
Tweet Processing
Offline Modules
Processed Tweets
Query (Entity)
Episode detection
Online Modules
Sentiment Analyzing
Result
1) All Episodes in the lifespan of the given entity. 2) For each episode, its most frequent nouns, polarity score. 3) Various Sentiment trend charts of the tweets related to the given entity query.

Figure 1: Flow Chart to detect episodes from Tweets using Episode Detection Algorithm

3.2.1 Tweet Collection Module

Tweet Collection module collects tweets using *Twitter Streaming API*. A sample of public tweets are extracted from twitter.com every 2 minutes. We have been collecting tweets since March 2012 and until December 2012. Around *140 Million* public tweets were collected from Twitter. Tweets were collected on an hourly basis; tweets for each hour are stored in a separate file.

3.2.2 Tweet Processing Module

Tweet processing includes removing non-english tweets and tweets with incomplete details. These processed tweets are stored by indexing them using Lucene [1]. The details about a tweet that are being stored in the Lucene index are tweet id, text, retweet count of that particular tweet and its creation time. In addition to this, the id, name, location, url, description, followers count, creation time of the account of the user who has tweeted the tweet are also stored for each tweet.

3.2.3 Episode Detection Module

A query(entity) is given as input to this module along with the processed Lucene Index from the above module. Episode detection module will extract all the tweets that are related to the given query and then all the episodes that have occurred over the life time of the entity are detected by applying Episode Detection Algorithm on the related tweets.

3.2.4 Sentiment Analysing Module

Sentiment Analysis is a method of analyzing/finding the opinion/sentiment that is expressed in a piece of text, a tweet in our context. In this module, a very basic sentiment scoring algorithm is applied on the tweets which are related to the given entity to get their sentiment score. This algorithm could be replaced with any other sentiment scoring algorithm; for this paper, we used a basic scoring algorithm as explained in Section 4.3. This module generates charts/graphs which shows how the sentiment of the entity has been changing over the period of its twitter lifetime.

We have given "Federer" query for our system along with the output of Tweet Collection and Tweet Processing offline modules and the flow is as below: (i) we retrieved episodes mentioned in Table 1 using Episode detection module, (ii) from episodes - we merged episodes and got bubble chart, (iii) we extracted sentiment scores and the trending graphs using sentiment analysis module.

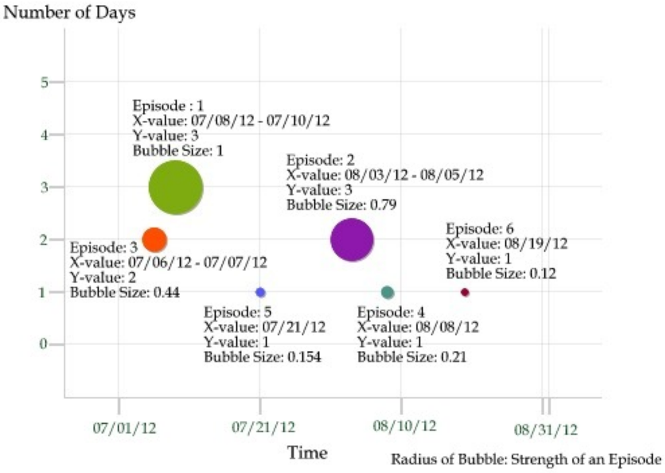


Figure 2: Episodes strength chart of entity "Federer" (see Equation 5)

4. TWEET EPISODE ANALYTICS

In this section, we present our algorithm to detect episodes from the tweet data. After the episode detection algorithm is executed on the data set, we use the information obtained from the algorithm to detect all the episodes of a particular entity. We also present sentiment analysis method that we have used in our system. In the post processing phase, we present sentiment, trend and temporal analytics of each episode.

4.1 Episode Detection Algorithm

Given an entity/query as an input, Episode Detection Algorithm gives episodes for an entity over a given time period. The algorithm will detect the episodes that have occurred in the entity's twitter lifetime. The time of birth for an entity in our twitter data set is the time stamp of the first occurring tweet that mentions it. Lifetime of an entity would be the first time stamp to till date. For this, all the tweets related to a given query are extracted from the Lucene index and are processed by cleaning the text. The proper nouns that have occurred in these tweets are determined using Stanford POS tagger[3] along with their frequency of occurrence in the tweets. Frequent bi-gram nouns are also extracted and then using the episode detection algorithm, all the episodes that have occurred over the lifetime of the entity are detected.

The following are the conditions to be satisfied to say that an episode has occurred on a short duration of time:

Table 1: Episodes detected of ‘Federer’

Rank	Episode	Date/Duration	Maximum Frequent Tweet [[Frequent Nouns]]	Frequency [[Tweet Spike]]	*Related Web URL [1]
3	Entering into Wimbledon ’12 finals	07/06/12 to 07/07/12	RT @Wimbledon: Federer will get a crack at his 7th #Wimbledon title beating Djokovic 6-3 3-6 6-4 6-3 to reach Sunday’s final. http://t.c ... [[Wimbledon, Federer, crack, Djokovic, title, Sunday]]	3464 [[22094]]	http://www.bbc.co.uk/sport/0/tennis/18740443
1	Winning Wimbledon ’12 title	07/08/12 to 07/10/12	RT @AndrewBloch: In 2003 a man predicted Federer would win 7 Wimbledon titles. He died in 2009 and left the bet to charity. Today Oxfam ... [[Federer, Wimbledon, title, man, Murray, today, bet, charity]]	6230 [[48919]]	http://www.atpworldtour.com/News/Tennis/2012/07/27/Wimbledon-Sunday2-Final-Report.aspx
5	Blog on Murray and Federer in Finals	07/21/12	RT @CrowdedSounds: Fan of both Federer and Murray? http://t.co/eOeQjSbu [[Fan, Federer, Murray]]	1636 [[7693]]	http://t.co/eOeQjSbu
2	About Federer	08/03/12 to 08/05/12	RT @Persie_Official: Federer is the boss [[Federer, gold, Andy, Murray, Wimbledon, mens, singles]]	3360 [[39646]]	–
4	Federer’s Birthday	08/08/12	RT @ATPWorldTour: Roger #Federer turns 31 today! Retweet to wish him a happy birthday! #atp #tennis [[Federer, Roger, Birthday, Today, retweet]]	2180 [[10832]]	http://www.tennisnow.com/News/Happy-Birthday-Mr–Federer.aspx
6	Winning Cincy Tennis title	08/19/12	RT @ATPWorldTour: #Federer beats @DjokerNole 60 76(7) to win fifth @CincyTennis crown, ties @RafaelNadala’s record 21 Masters 1000 titles ... [[Roger, Federer, Cincinnati, Masters, title, congrats, today, Djokovic]]	722 [[6022]]	http://www.espn.co.uk/tennis/sport/story/165924.html

1) The total number of tweets that are related to the event considering retweet count should be greater than *minNumTweets* (parameter).

$$T_E >= minNumTweets \tag{1}$$

where T_E is the total number of tweets that are related to event E.

2) For each day, spike extent (*spikeExtent*) is calculated. Let the day be represented by d and D is the number of days in the lifetime of given entity. The number of tweets related to the event E on a day d are *NumTweets(d,E)*

$$spikeExtent(d,E) = NumTweets(d,E) - NumTweets(d-1,E) \tag{2}$$

$$\max_{d=0}^{d=D}(spikeExtent(d,E)) >= spikeLimit \tag{3}$$

whereas

$$spikeLimit = T_E/spikeFactor \tag{4}$$

spikeFactor ($0 <$ *spikeFactor* $<= T_E$) is set manually. The maximum *spikeExtent* of all days should be greater than the *spikeLimit* threshold. The number of days the *spikeExtent* is greater than the *spikeLimit* is also counted as *spikeFreq*. The day on which the *spikeExtent* is maximum is the *spikeDay*.

3) The tweets on *spikeDay* are processed and then all the nouns in those tweets are extracted along with their occurrence frequency in the tweets. If the maximum frequent nouns which are most frequent after the query words corresponds to a single or at most two topics then the event is an *Episode*.

The difference between the number of tweets on a particular day and the number of tweets of the previous day is calculated for each day and the days are sorted in decreasing order based on this difference that is computed. The days which also satisfy the above conditions are considered as *spikeDays*.

The following additional information is extracted for each episode:

1) Let *Freq*$_N$, *Freqrt*$_N$ are arrays of nouns which are stored in decreasing order of their frequency from the tweets without and with retweet count correspondingly on the *spikeDay*. First 20 elements of *Freq*$_N$ and *Freqrt*$_N$ are extracted.

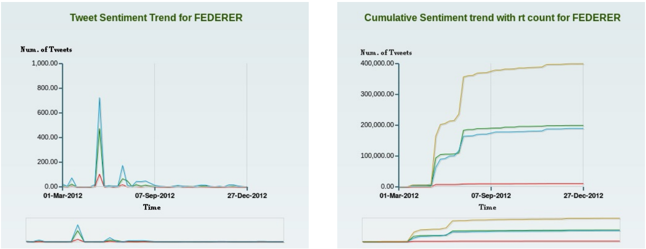
2) Let *Freq*$_B$, *Freqrt*$_B$ are arrays of bigram nouns which are stored in the decreasing order of their frequency from

[1] Note: * - not generated from our algorithm, but provided by us as a verification of the episode detected.

the tweets without and with retweet count correspondingly on the *spikeDay*. First 50 elements of $Freq_B$ and $Freqrt_B$ are extracted. Similarly, let us say $FreqPos_B$, $FreqNeg_B$ and $FreqNeu_B$ are arrays with bigrams which are extracted from tweets with positive, negative and neutral sentiments on the *spikeDay* correspondingly. First 50 elements from each of $FreqPos_B$, $FreqNeg_B$, $FreqNeu_B$ are also extracted.

3) Let *Tmax* is the tweet which has maximum retweet count on the *spikeDay* and *Tnoun* is array of nouns present in *Tmax*. *Tmax* is extracted and *Tnoun* is determined from *Tmax*. In addition to the above, the difference between maximum retweet count and minimum retweet count of the tweet on the *spikeDay* (*MaxMindiff*) is also extracted.

Figures 3.(a), 3.(b): Sentiment Trends of 'Federer'



From the tweets, all the above information is extracted and then top k (can be set manually) of the nouns, bigrams and the maximum frequent tweet, nouns in that tweet are all presented in the results as episodes.

4.2 Episode Analytics on Tweets

As a part of episode analytics for twitter, the sentiment trend and cumulative trend of tweets with retweet count are also presented as charts. Number of tweets with different polarities in each 100 tweets are also shown. For all the episodes their strength is calculated and presented in a chart. A chart with all the episodes of entity is generated and presented.

For an entity that has been given as input, until a maximum of 10 episodes are detected based on the threshold and the number of tweets related to the entity. The episodes are ranked based on their strengths. The strength of an episode is calculated as the ratio of the number of tweets that are tweeted about it and the time period over which the episode has occurred. The strength is the average number of tweets that are tweeted per day in the duration of the episode. The formula of the strength is given below:

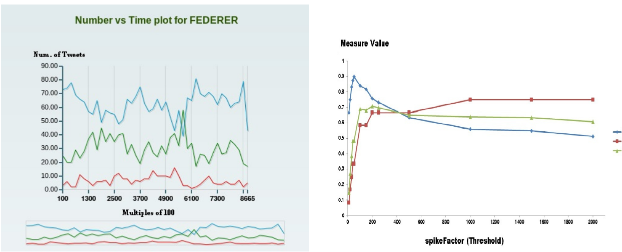
$$S_E = (\sum_{i=1}^{n} N_i)/n \tag{5}$$

where S_E is the Strength of an Episode (E) and N_i is the number of tweets on i^{th} day where as n is the number of days the episode has occurred.

The episodes are further sorted based on the time of their occurrence and all the episodes are presented from the start to the end of the lifetime of the entity. For us, the start and end times are the start and end points of the tweet collection.

Apart from the episode detection, the trends or patterns in the number of tweets and their sentiments are visualized. Basic polarity scoring algorithm is implemented by using cumulative polarity of adjectives. It is explained below in brief.

Figure 4.(a): Sentiment Trends of 'Federer' and Figure 4.(b): Thresholds measures chart



4.3 Sentiment Analysis

Given a piece of text, sentiment analysis algorithm will give the sentiment score of the text. The text is split by sentence and then all the words like stop words and others that has no sentiment or opinion in it are removed. The list of stop words used is taken from the Stanford stop word list[4] Sentiment lexicon has a list of words with their polarity score. It is taken from MPQA Subjectivity Lexicon[2]. The polarity score of the remaining words from the sentence which are present in the sentiment lexicon are added, which adds upto polarity score of a sentence. The polarity scores of all the sentences in the text are added to get the sentiment score of the total text. The sentiment score can be either positive, zero or negative, depending upon whether the text has positive opinion, neutral opinion or negative opinion.

5. RESULTS AND EVALUATION

In this section, we evaluate the proposed episode detection method by analysing the episodes strength for some famous personalities(entities). We have considered the twitter data from March 2012 to December 2012 for our experiments, so the episodes detected will fall into this timeline.

We have experimented with some queries like "Federer", "Serena Williams", "Lumia 920". We will be analysing the results on the entity query "Federer" in this section. Our Episode Detection algorithm has found 6 episodes related to "Federer" over the period of consideration(March '12 to December '12) and they are presented in Table 1 in sorted order of time.

Each Episode in the table has the following fields: Rank of the episode, episode description, date/duration of the episode, Maximum Frequent Tweet during the episode and Frequent Nouns, Frequency of the maximum frequent tweet and tweet spike, finally the web URL which shows details of the episode on the internet.

The rank of the episode is decided based on the strength of the episode that is being calculated. Episode description is the description in short for the episode that is detected. Date/duration of an episode is the period in which the episode has occurred. Maximum Frequent Tweet is the tweet which have occurred maximum number of times in the episode time period and Frequent Nouns are the nouns that are related to the episode which are sorted based on their frequency of occurrence. Frequency is the number of times the tweet has occurred where as tweet spike is the total number of tweets that are tweeted in the duration of the episode. For evaluating the episode that is detected, we have searched on the internet and then included the web URL of the page which shows the details of an episode and so proving the occurrence of that corresponding episode. Observe that the dates of the articles in the web URLs are same as the dates of occurrence of its corresponding episode. Each of the episode detected related to “Federer” is analysed further based on their date of occurrence below:

1) **The first episode** has occurred on 6th and 7th of July 2012 when Federer won the semi finals against Djokovic and entered into Wimbledon ’12 Finals just before the day of the finals. The rank of this episode is 3 and the maximum frequent tweet has tweeted 3464 times. The frequent nouns are wimbledon, federer, crack, Djokovic, title, sunday. The web URL shows that Federer has entered into finals by winning over Djokovic dated 6th of July 2012.

2) **The second episode** is after Federer winning the Wimbledon ’12 Finals over Murray. This episode is ranked number 1 and has occurred between 8th and 10th July 2012. Maximum frequent tweet has been tweeted 6230 times. Federer, wimbledon, title, man, murray, today are frequent nouns. The web page talks about Federer winning Wimbledon for the 7th time.

3) **The third episode** is the blog that is written about the final match between Federer and Murray and how people want both to win the match. This episode has occurred on 21st July 2012, 9days after the blog has been posted. Frequent nouns are fan, federer, murray. This might be because this is not an event, but the opinion of a person written in the form of a blog and so it took time to tweak. It is number 5 episode and the tweet itself has the URL to the blog.

4) **Robin Van Persie** tweets about Federer. Many people have retweeted it as they share the same opinion and so this has become an episode. The rank is 2 and this tweet has retweeted 3360 times. Federer, gold, Andy, Murray, wimbledon, mens, singles are frequent nouns.

5) **Federer’s 31st Birthday** is the fifth episode that has occurred on his birthday 8th August 2012. It is rank 4 and 2180 people has tweeted the same birthday wishes tweet to “Federer”. Frequent nouns are Roger, Federer, birthday, today.

6) **The last episode** is about Federer winning the Cincy Tennis Crown on 19th August 2012. Frequent nouns are Roger, Federer, Cinnicati, masters, title, congrats, today. The episode is ranked 6 and the url shows details about the episode.

All these episodes are sorted and their strengths are calculated and then the episodes strength of the entity is generated. The chart in figure 2 shows the strength of detected episodes of “Federer” with Time on X-axis and Number of days an episode has occurred on Y-axis. The radius of the bubble is taken as the strength of an episode. The strength is divided by 50000 to mark it as radius just to scale the value to fit into the chart.

Figures 3.(a), 3.(b) and 4.(a) shows the sentiment trends of tweets related to “Federer” over the time line. The sentiment trends charts are generated using Zingchart javascript library[5](free branded version). Figure 3.(a) shows the number of tweets that are tweeted positive (green line), negative (red line) or neutral (blue line) with sentiment on each day. Figure 3.(b) shows the number of tweets that are tweeted positive (green line), negative (red line), neutral (blue line) with sentiment or all in total (yellow line) until that day from the start day with retweet count. We can see there is a sudden spike in the number of tweets at several places. Figure 4.(a) shows the number of positive (green line), negative (red line) and neutral (blue line) tweets with sentiment that are present in every 100 tweets.

The episodes of “Narendra Modi” were also detected. “Narendra Modi” is an Indian Politician, Chief Minister of the state Gujarat in India. Table 2 shows episodes detected for the entity “Narendra Modi” with 6 episodes presented based on their occurrence date.

A brief analyis of the episodes detected is done below based on their date of occurrence: 1) The rank of the first episode is 1 and it occurred on 03/17/12. The episode is Modi on cover page of Time Magazine. 2) This episode occurred on 07/24/12 about Modi going to Japan. The rank of the episode is 3. 3) This episode is Modi wishing everyone on Janmastami. The rank of this episode is 4 and occurred on 08/10/12. 4) The episode with rank 6 has occurred on Modi’s Birthday on 09/17/12. 5) The episode occurred after Modi completed 4000 days as Gujarat’s CM and the rank of the episode is 5. It has occurred on 09/18/12. 6) Message from Modi is the next episode whose rank is 2. It has occurred on 10/13/12.

As a part of TEA system evaluation, we have calculated precision, recall and F-measure of our TEA approach. For an entity, the detected episodes are classified manually to be either valid or invalid episodes. An episode is valid if it is a significant event that has occurred in the lifespan of that particular entity. The ratio of number of episodes that are valid to the total number of episodes detected will be the precision of our TEA algorithm for that particular entity. The precision of TEA system is calculated by taking the average precision of all the entities.

The recall of TEA system for a particular entity is the ratio of number of valid episodes to the actual number of episodes that have occurred over that entity’s lifespan in twitter. The recall of our TEA algorithm is the average recall of all the entities. However, it is difficult to determine how many episodes have actually occurred for an entity over its twitter lifespan. So, for each entity we have manually searched over the internet (mostly their Wikipedia pages) and listed down the significant events that have occurred over a period from March 2012 to December 2012.

Table 3 shows the precision and recall for each entity that is given as input to the TEA system. The overall precision of the system that is calculated over these 11 entities is 0.864 whereas the overall recall of the system is 0.503.

F1-score (F-measure) is a measure of a test’s accuracy. The F1-score can be interpreted as a weighted average of the precision and recall and it’s formula is given by:

$$\textit{F1-score} = 2 * (\textit{Precision} * \textit{Recall})/(\textit{Precision} + \textit{recall}) \tag{6}$$

Table 4 shows the F1-score (f-measure) that are computed using precision and recall values from table 3 for each of the entities that are considered. The overall F1 score of TEA system is 0.62.

The precision, recall and f-measure values that are presented for different entities are calculated by setting different thresholds (*spikeFactor*) for different entities. These validation measure values change based on the threshold value that is set. For entity ‘Narendra Modi’, we have presented values of validation measures for different thresholds. Fig-

Table 2: Episodes of 'Narendra Modi' over its lifespan

Rank	Episode	Date/ Duration	Maximum Frequent Tweet [[Frequent Nouns]] [[Polarity Score]]	Frequency	*Related Web URL [2]
1	Modi on cover page of Times Magazine	03/17/12	RT @vijsimha: Here's news more interesting than #Budget2012. Time magazine puts Narendra Modi on cover as the man who could change Indi ... [[news, time, magazine, narendra, modi, cover, man]] [[1]]	314	http://timesofindia.indiatimes.com/india/Narendra-Modi-on-Time-magazine-cover/articleshow/12296366.cms
3	Modi going to Japan	07/24/12	RT @sardesairajdeep: Appreciate Narendra Modi for going to Japan and standing by Haryana govt. Nation above politics. (there you go folk ... [[narendra, modi, japan, standing, haryana, govt]] [[1]]	280	http://articles.economictimes.indiatimes.com/2012-07-23/news/32804624_1_maruti-suzuki-s-manesar-manesar-plant-maruti-s-manesar
4	Modi wishes on Janmashtami	08/10/12	RT @TOIBlogs: Janmashtami the protector of cows, Lord Krishna's birthday : Narendra Modi http://t.co/foHZ8Qwb [[protector, cow, lord, krishna, birthday, narendra]] [[1]]	86	http://t.co/foHZ8Qwb
6	Modi's Birthday	09/17/12	RT @Ohfakenews: Narendra Modi turns 62 today. You may remember him from his biggest hit: Naroda Patiya riots. #HappyBdayNamo #NaMo [[narendra, modi, today, hit, #happybdaynamo, #namo]] [[0]]	27	http://en.wikipedia.org/wiki/Narendra_Modi
5	4000 days as Gujarat's CM	09/18/12	RT @sardesairajdeep: Narendra Modi completes 4000 days as Gujarat chief minister today. Quite an achievement Shouldn't that be trending? [[narendra, modi, days, gujarat, chief, minister, today]] [[0]]	93	http://samvada.org/2012/news/4000-days-as-cm-narendra-modi-takes-gujarat-as-model-state-of-india-in-development/
2	Message from Modi	10/13/12	RT @Swamy39: Narendra Modi: UK has melted. US is not far behind. The hidden message is that if we are strong then they will come looking ... [[narendra, modi, message]] [[1]]	437	-

Table 3: Precision and Recall of Entities

Entity (query)	Precision	Recall	Entity (query)	Precision	Recall
Narendra Modi	0.9	0.333	Federer	1	0.588
Barack Obama	0.9	0.642	Britney Spears	0.8	0.4
Sachin	1	0.5	Serena Williams	1	0.83
Adele	0.5	0.5	Andy Murray	0.7	0.571
Life of Pi	0.9	0.33	Lumia 920	1	0.33
Taylor Swift	0.8	0.5			

Table 4: F-measure values of Entities

Entity (query)	F1 score	Entity (query)	F1 score
Narendra Modi	0.486	Federer	0.740
Barack Obama	0.749	Britney Spears	0.533
Sachin	0.667	Serena Williams	0.907
Adele	0.5	Andy Murray	0.629
Life of Pi	0.486	Lumia 920	0.499
Taylor Swift	0.615		

ure 4.(b) shows how precision, recall and f-measure values change with *spikeFactor* (threshold). The plot shows precision, recall and f-measure values on Y-axis for different thresholds on X-axis. The blue line in the plot corresponds to precision, maroon line corresponds to recall and green line to F-measure. The precision started low, increased to a maximum value and then decreased with increase in *spikeFactor*. Whereas, the recall started even low and increased

[2] Note: * - not generated from our algorithm, but provided by us as a verification of the episode detected.

Table 5: Validation measures for different thresholds of ‘Narendra Modi’

spikeFactor (Threshold)	Number of Episodes Detected	Precision	Recall	F1 score
10	3	0.67	0.08	0.15
20	4	0.75	0.17	0.27
30	6	0.83	0.25	0.38
40	9	0.88	0.33	0.48
50	9	0.9	0.33	0.49
100	20	0.84	0.58	0.69
150	23	0.82	0.58	0.68
200	30	0.76	0.67	0.71
250	39	0.73	0.67	0.7
500	61	0.63	0.67	0.65
1000	85	0.56	0.75	0.64
1500	105	0.55	0.75	0.63
2000	120	0.51	0.75	0.61

with *spikeFactor* until it reached a maximum value and then it became constant from there. F-measure followed a similar pattern as that of precision curve. Table 5 shows the precision, recall and f-measure values for different *spikeFactor*.

The top 6 episodes that are detected for entity ‘Narendra Modi’ when threshold (*spikeFactor*) is set to be 50 are presented in Table 2 and validation measures for different thresholds for ‘Narendra Modi’ are presented in Table 5.

6. CONCLUSIONS

Our intention to infer significant knowledge/insight from huge number of tweets raises problems. The key issue is to comprehend what a set of tweets convey about an entity. Our approach has been to consider lifetime of an entity and determine what all events can occur in it. From the events one can get episodes that convey larger description of the set of tweets are conveying, and then episodes strength of an entity are shown. We built a system for taking any entity as a keyword and process relevant tweets to detect the episodes. Our results validate our approach by providing episodes that provide the essence of information that can be gleaned from tweets. In particular, we are able to convey sentiments about tweets and phrases that describe tweets over different periods of time. Therefore, our system can be used to determine short term understanding from tweets about a given entity and use it to promote or rectify certain actions. For example, sell more mobile phones at discount or quickly send out a patch for a malfunctioning applet. As part of future work we will continue to improve core algorithms applied in this paper, and delve into what can be learned from detected episodes.

References

[1] *Apache Lucene.* https://lucene.apache.org/.

[2] *MPQA Subjectivity Lexicon.* http://mpqa.cs.pitt.edu/.

[3] *Stanford Part-Of-Speech Tagger.* http://nlp.stanford.edu/software/tagger.shtml.

[4] *Stanford Stop-Word List.* http://www.wordsift.com/wordlists.

[5] *ZingChart Javascript Charting Library.* http://www.zingchart.com.

[6] S. Asur and B. A. Huberman. Trends in social media: Persistence and decay. *AAAI*, 2011.

[7] H. Becker and M. Naaman. Beyond trending topics: Real-world event identification on twitter. *AAAI*, 2011.

[8] M. S. Bernstein and B. S. Eddi. Interactive topic-based browsing of social status streams. *UIST*, 2010.

[9] D. Gruhl and R. Guha. Information diffusion through blogspace. *WWW*, 2004.

[10] M. Iwata and T. Saka. Aspectiles: Tile-based visualization of diversified web search results. *SIGIR*, 2012.

[11] H. Kwak and C. Lee. What is twitter, a social network or a news media? *WWW*, 2010.

[12] M. Mathioudakis and N. Koudas. Twittermonitor: Trend detection over the twitter stream. *SIGMOD*, 2010.

[13] M. R. Morris and S. Counts. Tweeting is believing? understanding microblog credibility perceptions. *CSCW*, 2012.

[14] J. Nichols and J. Mahmud. Summarizing sporting events using twitter. *ACM IUI*, 2012.

[15] T. Sakaki and M. Okazaki. Earthquake shakes twitter users: real-time event detection by social sensors. *WWW*, 2010.

[16] Teevan and Ramage. Twittersearch: A comparison of microblog search and web search. *WSDM*, 2011.