

## REAL-TIME TWITTER TREND MINING

V.Shyam Kumar<sup>1</sup>, G.Varshini<sup>2</sup>,G.Akanksha<sup>3</sup>,I.Sravani<sup>4</sup>,S.Shirisha<sup>5</sup>

1Assistant Professor, Department Of CSE., Malla Reddy College Of Engineering For Women.,

Maisammaguda.,Medchal., Ts, India (✉vsmca15@gmail.Com)

2, 3, 4, 5 B.TechCSE, (19RG1A0571, 19RG1A0572, 19RG1A0577, 19RG1A05B4),

Malla Reddy College Of Engineering For Women., Maisammaguda., Medchal., Ts, India

### ABSTRACT

*One of the major sources of trending news, events and opinion in the current age is micro blogging. Twitter, being one of them, is extensively used to mine data about public responses and event updates. This paper intends to propose methods to filter tweets to obtain the most accurately descriptive tweets, which communicates the content of the trend. It also potentially ranks the tweets according to relevance. The principle behind the ranking mechanism would be the assumed tendencies in the natural language used by the users. The mapping frequencies of occurrence of words and related hash tags is used to create a weighted score for each tweet in the sample space obtained from twitter on a particular trend.*

### 1. INTRODUCTION

Trend Analysis is spotting a pattern of occurrence or tendencies that data or information can portray. Trend analysis is often used to predict future events, like the likeability of a product, or the viewership of a match, etc [1]. Micro blogging is a medium of broadcast which is similar to blogging. The aggregate and the actual blog size are typically smaller than a traditional blog. Micro blogs allow users to share content such as short sentences, individual images or video links [2]. Twitter is one of the most popular micro blogging websites on the Internet today. It uses 'Hash tags', which is Hash '#', followed by the trend-name, to identify and group micro-posts about a particular trend. Micro-posts in twitter are known as 'tweets'. Twitter also categorizes tweets and trends according to region and on its web interface it displays the most recent tweets as feeds [3]. Twitter also provides handles in the form of API for developers to create applications centered on the data gathered by tweets and trends [4].

The data gathered by the tweets is generally unprocessed and to determine any significant conclusions from it requires it to be mined and processed. The Hash tags generally are telegram style word(s) description without spaces and attempts to summarize the trend, but seldom is a trend completely understood by its hash tag. The methods Proposed in this paper takes sample data from twitter for the most trending Hash tags and attempts to summarize the trend by displaying the most descriptive tweets which can summarize the content of the trend. It also prescribes a weighted score to the sampled tweets in order to rank it according to accuracy of its description in its content about the Hashtag. This would enable a better informing feed about the trend than a 'recent' based display currently used in the web interface.

### 2. ASSUMPTIONS & SCOPE OF WORK

It is assumed that the tweets in themselves are not highly descriptive. One of the constraints that Twitter allows a maximum of 140 characters, alphanumeric and special characters that can be used in a tweet. This constraint makes tweets be in a telegram format where key words are prioritized over grammatical correctness. The scope of the methods proposed in this paper extends to assign scores to the tweets taken as sample space so as to be able to rank them as highest descriptive with the highest score. This can be used to accustom a new user to get acquainted with the trend content.

### 3. TWEET ANALYSYS ALGORITHM

#### 3.1 Description of the algorithm

The algorithm takes as input a sample space of 'n' predefined number of tweets. It also takes the highest trending 'x' number of trends. The output of running the algorithm is the tweets of the sample space

ranked with a decreasing description index. The algorithm also uses two dictionaries. The first dictionary contains the list of the words, which have less significance to the content description and are more grammatical tools, namely articles, prepositions and conjunctions. The second dictionary consists of all common nouns, adjectives, adverbs, verbs and their derivatives. The former will be called 'filter' and the latter 'centiliter' hereon.

The used sample space is placed in a file, separated by an end of tweet character, like '%%'. Once the tweets are acquired, the frequency of every word that is used in the file containing the tweet sample space is found. This would exclude the '#' tags and the '@' tags. The URLs in the tweets are also ignored while finding the frequencies. Hence the list of words and their corresponding frequencies is prepared and stored. It is now to check for association of the highest trending tweet with the other high trending tweets. Tweets about the same event, or person, hold useful content and can be assumed to contain more relevant data. The tweets with a high trending hashtag along with the highest trending hashtag for the second time are used to collect the frequency so as to update the previously generated frequency table. Once the frequency list is obtained we perform a rating on the words to find its weighted score. This weighted score is used to get the cumulative score of each tweet, which can be used to rank the tweets according to its content relevance. Also proposed is a way to learn from the newer tweets about the hashtag and get more accurate tweet ranking.

### 3.2 Sample space preparation

The first requirement is to find a sample space for the algorithm to run. This finite sample space is a subset of the data that is available at twitter. For the purpose of the explanation, the number of tweets taken in the sample space 'SS' is 'n'. This number 'n' when increased will improve the accuracy of the analysis but also increase the average runtime, but when decreased it radically decreases the accuracy of the findings. When writing the SS into a file, it has to be made sure of that either an end of tweet character logically separates them or the file format is such that individual tweets can be logically separated. As is used in the algorithm, the top trending 'x' hashtags are also along with the SS.

---

```
For tweet in retrived_tweets
```

```
  If tweet has highest_tag
```

```
    Add tweet to SS
```

```
  // Adding again into SS
```

```
  For tag in x
```

```
    If tweet has tag
```

```
      Add tweet to SS
```

---

### 3.3 Ranking algorithm

The Algorithm is run in two stages. The first stage is the word frequency index and the second stage is tweet rating.

#### 3.3.1 Frequency Index

The Frequency of the words that are used in the SS is used as the fundamental data for content rating. The Algorithm for calculating frequency index is as follows:

---

```
Word_count = 0
```

```
For word in SS
```

```
  If word in Freq_list
```

```
    Freq_list[word] = Freq_list[word] + 1
```

```
  Else
```

```
    Freq_list[word] = 1
```

```
  Word_count = word_count + 1
```

```
For word in Freq_list
```

```
  Freq_index[word] = Freq_list[word] / Word_count
```

---

#### 3.3.2 Tweet Score

The Tweet score is an index generated for each Tweet 'T' in the sample space 'SS' which will be the basis for the rankings. The generated rankings would be in the decreasing order of the Tweet score.

The Algorithm of for the generation of the Tweet score is as follows:

---

```

For Tweet in SS
  Score[Tweet] = 0
  For word in Tweet
    If word not in filter
      Score[Tweet] = Score[Tweet] + Freq_index[word]
    If word in cnfilter
      Cnindex[Tweet] = Cnindex[Tweet] + 1
  If Cnindex[Tweet] = word_count(Tweet)
    allCn[Tweet] = True
  Else allCn[Tweet] = False
    
```

---



---

```

Word_count = LI * Word_count
For word in new_SS
  If word not in new_Freq_list
    new_Freq_list[word] = (LI * Freq_list[word]) + 1
  Else
    new_Freq_list[word] = new_Freq_list[word] + 1
  Word_count = Word_count + 1
For word in new_Freq_list
  Freq_index[word] = Freq_list[word] / Word_count
    
```

---

### 3.3.3 Dynamic Learning

While generating the score for each tweet in the sample space, we are overlooking the newly broadcasted tweets made after the sample space collection. Hence, the generated score might become obsolete after a finite time. To accommodate dynamic learning in the algorithm we introduce a learning index ‘LI’ which determines the rate of decomposition of the previously collected data and the exclusive inclusion of the newer data. To determine the ‘LI’ we need to decide the decomposition rate. If we want the contribution of the old data to reduce to ‘t’ percent after ‘f’ new incoming tweet streams then the value of ‘LI’ can be calculated as,

$$t = n (LI)^f$$

$$LI = (t / n)^{(1/f)}$$

Once this Learning Index LI is calculated the Dynamic Algorithm can be stated as

## 4. GENERATED RESULTS

Since the frequency of words is the primary source of classification in this ranking algorithm, it is important to observe the distinguishing frequency. For the target high frequency words to contribute more towards the score of a Tweet its frequency should ideally be substantially more than the more common words, whose contribution should be insignificant. We can observe the trend in a sample run of the frequency index algorithm as follows.

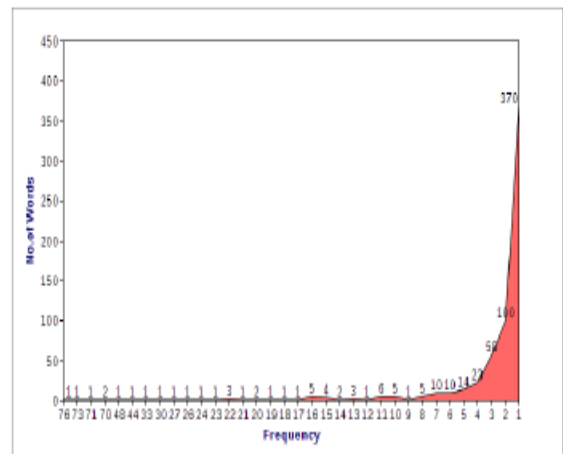


Figure 1. Plot of No. of words vs Frequency

As can be inferred, the 22 highest frequency words, among 549 words contribute approximately 56% of the Sample Space SS content. This shows the tendency of algorithms based on frequency of words to have a definite validity [5].

### 4.1 Example

On running the above algorithm, we get the following results,

**Table 1. Highest Trending:**

**#BollywoodEnglishTitles**

Word	Frequency
More	24
Trending	23
Stand	22
Total	22
Winners	22

In the above run the most eligible or the most highly ranked tweet, “So Tweeple, we’re trending at No. 1!!!2 more winners! So our contests’ total winners now stand at 7!!!” indicates that the trend is a contest where the winners are decided according to the highest trending. We can again see the high frequency word tendency to be different from common word usage [6]. In the above example the SS contained 500 tweets as a sample space. On running the algorithm in a different geographical, the following result was obtained,

**Table 2. Highest Trending: #Exam Moments**

Word	Frequency
Pass	12
Paper	12
Look	9
Moment	8
Know	6

In the above run the most eligible run returned as the most eligible tweet to be “A: Where did you get your answer? B: Everybody looks to the left, everybody looks to the right #Exam Moments”, which indicates towards a more general phenomenon. In both the runs the 10 most used word’s occurrence contributed to an average of about 51% with a standard deviation of 3%. These statistics, by induction, validates our assumption that the frequency of words can govern

the descriptive quality of a tweet. It can also be logically concluded that the tweets fairly depicts the essence of the event, though there is no quantitative method to numerically analyze that.

## 5. FUTURE IMPROVEMENTS

In the current implementation we do not consider the length of the tweet. As per restrictions, a tweet may not exceed 140 characters, but when we run the proposed algorithms we do not consider whether the tweet is using the maximum available character space. One way to do it is to calculate the Tweet score in the following manner,

$$Tweet\_score = \left[ \sum word\_score \right] * (word\_count \div 140)$$

In this way, the Tweet that is most highly ranked is considered to be using the maximum character space available. The results generated after using this enhancement is beyond the scope of this literature.

## 6. CONCLUSION

Considering the results generated by the algorithms produced by this study, the Tweets can successfully be used to describe the content of the trends. Since the ranking algorithms logically hold true for most of the tweet types, it is still not a good way to detect spam or RT, ReTweets [7]. For implementing those features Image and URL checking algorithms needs to be in place. Also these algorithms do not take into considerations the ‘favorite’ attribute that is associated with each tweet. A ‘favorite’ attribute is a counter which is incremented whenever a user up votes a particular tweet [8][9].

## 7. REFERENCES

[1] D. Zhao and M. B. Rosson. How and why people twitter: the role that micro-blogging plays in informal communication at work. In *Proceedings of the ACM 2009 international conference on supporting group work*. ACM, 2009.

[2] boyd, danah, Scott Golder, and Gilad Lotan. 2010. "Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter." HICSS-43. IEEE: Kauai, HI, January 6

[3] Shirley Ann Williams, Melissa Terras, Claire Warwick (2013). "What people study when they study Twitter: Classifying Twitter related academic papers". *Journal of Documentation*, 69 (3).

[4] Twitter Search API.  
<http://apiwiki.twitter.com/Twitter-API-Documentation>.

[5] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proc. of the National Academy of Sciences*, 105(41):15649–15653, 2008.

[6] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005.

[7] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *Proc. of the 4th ACM European conference on Computer systems*. ACM, 2009.

[8] M. E. J. Newman and J. Park. Why social networks are different from other types of networks. *Phys. Rev. E*, 68(3):036122, Sep 2003.