Engineering and Technology Journal e-ISSN: 2456-3358
Volume 05 Issue 07 July-2020, Page No.-711-719
DOI:10.47191/etj/v5i7.02, I.F. – 6.127

A Study on Timeframe based Radical Event Detection on Online News Archive

Pritish Gupta[1], Prof. Sridhar Ranganathan
[1]SCOPE, Vellore Institute of Technology Chennai, Tamil Nadu, India
[2]Associate Professor, SCOPE Vellore Institute of Technology Chennai, Tamil Nadu, India

Abstract: In the field of Natural Language Processing, Event detection has been an active area of research. While most of the work emphasizes on detecting all possible events by using datasets comprising of news article belonging from a broad range of time and location (collected either from a single data source or multiple data sources), this work focuses on detecting an event using short timeframes of News Articles extracted from an Online News Archive. Also, this is a non-targeting approach when it comes to detecting the theme or category of event it focuses on but becomes a targeted one as it tries to put all its focus on detecting one major event when applied on a timeframe. While other approaches are based on various techniques like semantic graphs, clustering algorithms, topic detection and tracking for event detection, this approach detects events with the help of a filtering mechanism combined with a novel threshold. This algorithm leverages detection of trending characteristics of an event in a timeframe and filtering news articles which show similar characteristics. And finally, we evaluate our algorithm with precision, recall and the percentage of articles which are actually related to the event in a timeframe termed as dominance percentage.

I. INTRODUCTION

With time, our lives have become much faster paced than it used to be. Technology has a very big role in making it so. With everything coming to online platform nowadays, people are able to make the most of their time. There are a lot of things people have switched from offline to online mode. News has been no different. Digital viewership of news through the medium of Mobile apps, websites etc. has also increased a lot. But since people don't have a lot of time to go through each and every detail of News Events, it has become important to detect the most important/major event happening around. While there is a lot of existing literature for Event Detection, our work attempts to provide a new perspective to solving the problem. Our work makes use of timeframes in order to detect a timeframe unlike other existing work that focuses on detecting all the events within a given dataset of news articles belonging from a broad range of time.

II. RELATED LITERATURE

This section discusses about the recent literature in this field. The exiting literature discusses about two different approaches, targeted event detection and non-targeted event detection. In all these approaches, targeting an event refers to focusing on event that are from a particular domain/theme. Whereas non-targeted event refers to detecting generic events. In that context, our algorithm is a non-targeted one. But our algorithm focuses on detecting one major event being discussed in the given time frame.

Among the existing literature, the majority of work that has already been done exists in the **Non-targeted event detection** area. Various approaches have been introduced for identifying events that belong to a broad range of domains/themes/categories . J Allan et. al. [2], Y. Yang et. al. [3], T. Brants et.al. [4] presents methods which are variants of *TF-IDF* model. Some researchers of have proposed term level/based analysis.Fung et al.'s work is somewhat similar to our work in spirit but differs in concept. In [5] G. P. C. Fung et al. have proposed an approach that identifies bursts of term groups by taking into account frequency of terms as well as documents across which it is occurring and then have compared it with expected frequency to detect an event. Our approach differs from this approach as it focuses on detecting a single event in a timeframe, in the author's work terms can be very generic and are obtained from model which are very similar to *TF-IDF* model but our work is mainly based on superstring matching and then finding out trending characteristics instead of calculating the burstiness. Our approach also provides a lot of focus on the what characterizes an event and hence takes into consideration location and keywords whereas in the above work, items are generic. In [11] [12] [13], segment level approaches for event detection have been proposed. While Leskovec et. al. [11] uses memes and text segments, Li et. al. [12] work with tweets and consecutive n-gram to detect meaningful sentences.

The existing literature also consists of targeted methodologies which takes into consideration one specific theme when detecting any event. Although the algorithms are capable of detecting events from all backgrounds but while detecting, the algorithm targets the theme/background of the event being detected. This is known as **Targeted Event Detection**. Existing literature
on targeted event detection includes [1] [9] [10] [14] [15] [16] [17] [18]. Yifang Wei et. al. [1] proposes a graph-based method which utilizes a location ontology and a domain dictionary to identify news articles from large noisy corpus. Their proposed a system which detects events with the help of semantic graph creation. Another direction of research [14] [15] [16] is lexico-syntactic or lexico- semantic patterns to identify events. These approaches rely on finding specific patterns in events; however, in real world, a considerable portion of text associated with targeted events may differ from these patterns. This could result in a non-trivial miss rate. Wang et. al. [19] have proposed learning pattern from the event instead of relying on predefined patterns. Another set of approaches that are somewhat similar to our work in spirit are [9] [10] [17] [18]. There are some major differences in our approach. The approaches described in the existing literature are binary in nature and do not specifically detect an event but the final outcome of the approaches is to detect the presence of event whereas our work defines Event Detection in a way to filter out News Articles related to the event. The place where these approaches becomes similar to ours is the usage of timeframes or time window as referred in the existing literature. But the way of detecting events is completely different. These approaches [9] [10]
are typical domain specific approaches which start with a domain vocabulary collected by domain experts which are used to filter out raw corpus in a specific time window to signify the domain of events present. Whereas our work is a non-targeted approach which doesn't require any domain vocabulary and is solely based on collection trending information in a short frame of time where the probability of detecting a major event is high. Muthiah et. al. [27] proposes a different approach where instead of using keywords, they use seed patterns and then use bootstrapping strategy to learn more hidden patterns which are then used to identify documents (specifically tweets) relevant to target event. Our approach uses none of these strategies and rather than using tweets or messages by public (which can be difficult to put in context correctly) makes use of timeframe of News Articles leveraging the trend detection and novel filtering mechanism to detect the occurrence of any (domain independent) radical/major event.

III. DEFINITIONS AND ASSUMPTIONS

In this section we present definitions, assumptions, and problem statement.

Definitions: An **Event** is something that happens at a particular time and place which gets covered by various platforms. A **Radical Event** refers to any event that has a major impact on people such that News Agencies start giving a lot of coverage to this event. For e.g.- The Indian Lok Sabha Elections can be considered as one Radical Event. It is easily noticeable that when the counting of votes starts, media gets focused on this particular event. News Articles are written in humongous volumes about this event thus making this event a dominant event in that timeframe. Another example of a Radical Event can be when a terrorist attack happened in the Pulwama district of Jammu & Kashmir. Again, when the attack happened, News Agencies wrote a lot about different aspects of the event including what happened, how many soldiers got killed to involvement of Pakistan behind this attack etc. Radical Event refers to any event that is written about so much so that it becomes a substantial percentage of all the news present in that short timeframe.

Assumptions: A timeframe is a collection of news articles where it is assumed that the probability of detecting an event that the user is looking for is considerably high. A timeframe should consist of News articles which are related to at least one major event. A timeframe is also assumed to contain news articles of the day the event took place as well as the following days till the time the event is still being discussed. The timeframe is assumed to contain one major event that is being targeted. It is also assumed that at least an estimate time of occurrence of an event is known to create a good timeframe. According to out work, Event Detection is described as filtering out news articles that are related to the major/radical event.

Problem Statement: Given an estimate of the time of occurrence of any particular event and access to News Articles of any publication, the task of this algorithm is to detect the major event that is being targeted w.r.t time of its occurrence.

IV. EVENT DETECTION

In order to identify an event, the process of Event Detection is divided into three parts (i) Timeframe Creation (ii) Trending Characteristics detection (iii) Filtering related News Articles. The following is an overview of an algorithm for event detection.

Algorithm 1: Overview of Event Detection

Input:

A timeframe of News Articles: Tf

Output:

A Set of filtered news articles related to the event: FTf

Steps:

1. TL = identify_trending_locations(Tf)
2. TK = identify_trending_keywords(Tf)
3. For each article A:

Tl = identify_locations(A) Tk = identify_keywords(A)

4. Calculate locpercentage and keypercentage
5. Calculate relscore = locpercentage + keypercentage

6. If relscore (A) > threshold:
Append A to FTf
7. Return FTf

When a timeframe Tf is passed on to this algorithm (keeping in mind the timeframe should abide with all the assumptions as discussed), a Filtered timeframe is created which consists of the News Articles which according to the algorithm discusses about the major event that happened in that timeframe. The approach takes advantage of the short timeframe that is being focused upon and leverages the trending characteristics of the timeframe to detect an event. And then by matching and filtering all the News Articles in the timeframe, a filtered timeframe is created which is supposed to consist of News Articles that are actually related to the major event thus in a way providing us a way to understand the major event happening in the timeframe.

A. WHY TIMEFRAME AND TIMEFRAME CREATION

Whenever an event is covered by news articles, the news is only discussed for a specific period of time. Hence to detect an event, it is important to look for it in the correct period of time. Hence, if an estimate of when the event actually occurred is known, there is a higher chance for an algorithm to detect an event. Hence, for this reason Timeframe was considered.

For demonstration purposes, five events are taken from different backgrounds. The five events are as follows (i) Pulwama Attack (ii) Balakot Strike (iii) Scrapping of Article 370 (iv) Babri Masjid vs Ram Mandir Verdict and (v) CAA/CAB protest. All of these timeframes are from different background for e.g. Pulwama Attack and Balakot Strike are from violence/border tension, Article 370 and CAB/CAA are from political background and Ram Mandir vs Babri Masjid verdict is from judicial background. For creation of all of these timeframes, a timeframe of 2-4 days was taken. These timeframes are all general and have not been experimented with by excluding or including articles of certain days.

The timeframe created is just raw set of articles taken by considering two factors discussed in the assumption section. First, the timeframe includes articles from the date on which the event took place and Second, News Articles from a few following days are also taken. Articles from the days back can also be considered for testing the limits of the algorithm but have not been considered in detecting events from these timeframes. **Note-** The focus of the focuses on one more characteristic i.e. trending keywords of the timeframe. To detect trending locations and keywords that are being discussed in the timeframe, a superstring matching approach has been used. The algorithm for finding trending keywords and trending locations are as follows.

TRENDING LOCATION DETECTION

In finding locations in a text there is one major problem, many a times the same location that is being discussed is written in many different ways e.g. Jammu, Kashmir Jammu & Kashmir when mentioned in an article tend to represent roughly the same place or region. To counter this problem a novel superstring matching algorithm is used. This algorithm tries to make groups of such locations and represent it in the largest superstring location which overlaps all the other substrings. To detect locations in a text, spacy has been used. The pretrained entity recognition method in spacy is used to detect any kind of location or geo political entity and then grouping of these locations is done so that all the places in the text are grouped and are represented by the same superstring. The algorithm for superstring matching is as follows.

The Superstring matching algorithm first takes in all the pre-processed elements (location/keyword) collected from the timeframe and then attempts to group locations that roughly represents the same element. For this there are a total of 4 cases that two strings S1 and S2 detected in the timeframe as the elements can be in. (i) Both S1 and S2 are identical, (ii) S1 is a complete subset of S2, (iii) S2 is a complete subset of S1, (iv) S1 and S2 are different. This work doesn't take into consideration partial matching as then the chances of false matching becomes extremely high. For Superstring matching algorithm, each time an element e is picked from list of pre-processed elements, it is check with the elements in the dictionary created. There are 4 possibilities. (i) e is identical to an existing key in the dictionary, (ii) e is a subset of an existing key, (iii) e is a superset of an existing key or (iv) e doesn't exists in the list of keys in the dictionary. For each of these conditions a state is assigned. State 0 means e already exists as a key. State 1 means e is a subset of an existing key. State 2 means e is a superset of a single or multiple exiting key/keys. State 3 means e doesn't exist in the dictionary as a key algorithm is to detect the event, not efficient timeframe creation. It is assumed that the date at which the event occurred is known or some idea about the timing of the event is known such that articles from that time period (2- 4 days max) can be bundled together to form a timeframe.

B. TRENDING CHARACTERISTIC DETECION

An event is considered by something that happened at a given place and time but the existing literature doesn't provide any detail about what an event comprises of. One of the aims of this paper is to provide some insight into what can be considered as the characteristics of an event. Since for an event to be recognizable it has to happen at some place, location can be one of the basic characteristics which all the events comprise of. To detect event, this approach.

Algorithm 2: Superstring-Matching (Part 1).

Input:

A list of cleaned/pre-processed locations: L.

Output:

A dictionary containing the super-string location name as the key and its frequency count as the value: D.

Steps:

1. Initialize an empty dictionary: D
2. for location in L:

state, key = get_state_and_key(location) if state ==0 or state ==1:
D[key] +=1 else if state == 2:
D[location] = 1 for k in key:
D[loc] += D.pop(k)
else:
D[loc] = 1

3. return D

Algorithm 2: Get State and Key (Part two)

Input:

Element being checked: e. Dictionary of elements: D

Output:

State of the element being checked
The key in the dictionary, the element being checked will be represented by.

1. if e in D:

return (0, e)

2. else:
3. if e is a subset of keys in the dictionary: return (1, superset_key)
4. if e is a superset of single/multiple keys: return (2, list_of_subset_keys)
5. return (3, None)

This algorithm is being used in Location and Keyword detection algorithm to group locations and keywords representing roughly one thing. For location and keywords detection, pretrained methods of spacy have been used to detect entities and from these entities, l

Algorithm 3: Location Detection

Input:

A list of possible locations detected with spacy: PL.
A list of countries, capitals, their 2 lettered notations and 3 lettered notations collected with the help of pycountry: WL.

Output:

Top 5 trending locations in the timeframe: TL

Steps:

1. Convert all the location items in PL and WL to lowercase.
2. Strip each of the location and if there exists "the" in the location item, remove it.
3. RL = Filter out all the elements occurring in WL from PL if there exists any.
4. D = superstring_matching(RL)
5. TL = Filter out top 5 locations with the highest frequency.
6. Return TL

By detecting trending location, we are able to detect the venue of the event. Another characteristic which is considered to provide a good insight to the event being detected is keywords.

2. TRENDING KEYWORDS DETECTION

For extracting keywords, an approach similar to the one in detecting trending locations is used. The core of the algorithm still remains to be the superstring matching algorithm. The main difference in between the two algorithms is that instead of locations, keywords are being used and a bit more pre-processing is required to create a list of keywords.

Algorithm 4: Keywords Detection

Input:

A list of all the articles in timeframe: Tf

Output:

Top 5 trending locations in the timeframe: TK

Steps:

1. Initialize an empty list of possible keywords: PK
2. Convert the document content in each News Article $\boldsymbol{n \in Tf}$ to processed content using spacy.
3. for each token t in the processed content: if token.lemma_ != "-PRON-" and
token.pos_ not in ['AUX', 'ADJ', 'DET',
'VERB', 'NUM', 'ADV', 'PUNCT']: append t to PK
4. D = superstring_matching(PK)
5. for each keyword in D:

doc_freq = calc number of documents in which this term occurs.

6. Update the value corresponding to each keyword in D with TF*DF for each keyword
7. TK = Filter out top 5 keywords from the D.
8. Return TK

After detecting the trending locations and keywords, an insight can be gained on what the major event was about. The location provides a detail about the Event Venue whereas the keywords provide any slogans or words that are trending in the timeframe. Keywords provides an insight to what kind of opinions are formed in the minds of people.
The results of the location detection algorithm when applied to each of the timeframes are show as follows.

C. ALGORITHM OUTCOMES

1. RESULTS OF TRENDING LOCATION DETECTION

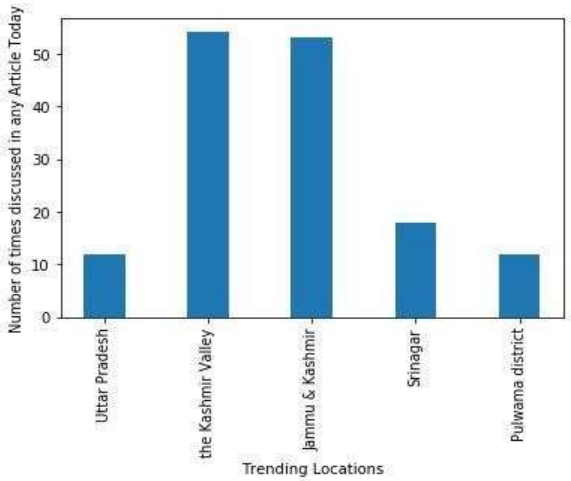

Figure 1: Timeframe 1 (Pulwama Attack)

The event which took place in Pulwama region of Jammu & Kashmir has been successfully detected. The algorithm has been successful in capturing the places affected most by the event by detecting Pulwama District, Jammu & Kashmir (along with the valley region), Srinagar. Since, the Chief Minister was very vocal about the event, the algorithm was also successful in detecting that too.

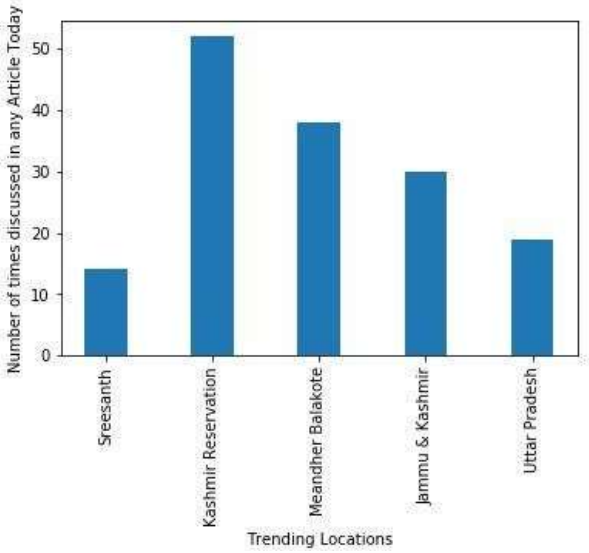

Figure 2: Timeframe 2 (Balakot Strike)

This event was a response of the previous event from India to Pakistan. An air-strike took place at the Balakot region of Pakistan in an attempt to blow up terrorist camps. The algorithm was successful in detecting Meandher Balakot, Kashmir was detected as a result of retaliation by Pakistan over Kashmir region. The algorithm also suggest that Uttar Pradesh was also an important location being discussed in that timeframe. The actual reason was the same as above.

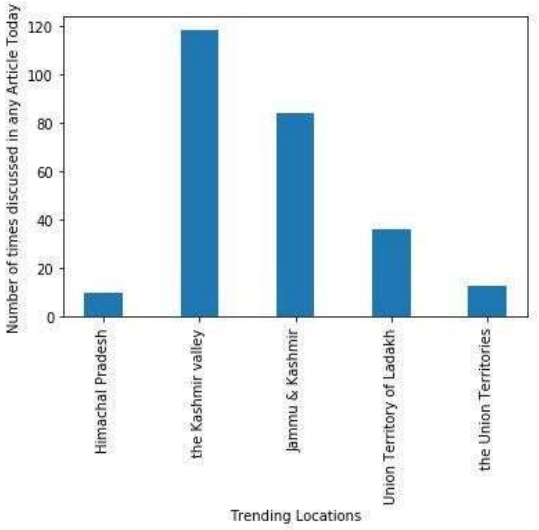

Figure 3: Timeframe 3 (Scrapping of Article 370)

When Article 370 which was being talked about the main reason behind Jammu & Kashmir not being developed a lot as compared to other states was scrapped, each part of Jammu & Kashmir state was divided into 3 Union Territories. The algorithm has successfully managed to catch all that information along with the location this event was most related to/had a direct impact on.

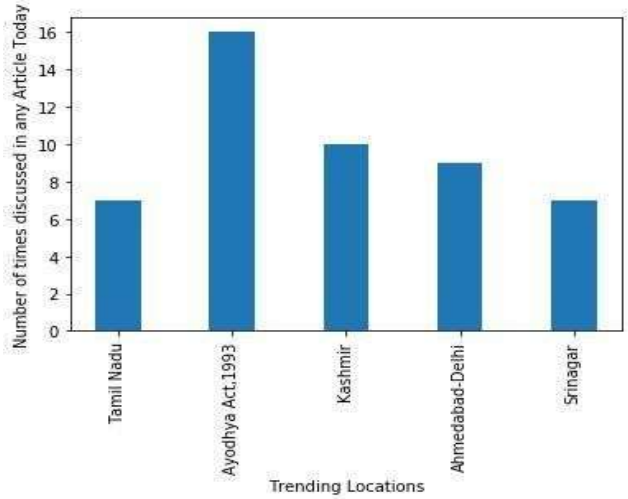

Figure 4: Timeframe 4 (Babri Masjid vs Ram Mandir Verdict)

Another event that happened was the verdict of the long pending case of land dispute between the followers of two religions. The disputed land once had Babri Masjid which was taken down by a mob and a case was registered claiming that the land belonged to Lord Ram as it was his birth place and was later forcefully taken over by the famous ruler Babar. The case was long pending in Supreme Court. The decision of this case came after a very long period of time. Since, India consists of people from both the religion in a considerable amount, the case became a major headline. The algorithm is successful in detecting Ayodhya as the location of disputed land. Since, Kashmir (including Srinagar) has a lot of Muslim

population, the sentimental implication of the verdict on Muslim communities of Kashmir was also discussed. People from Tamil Nadu had mixed opinions on the verdict by Supreme Court.

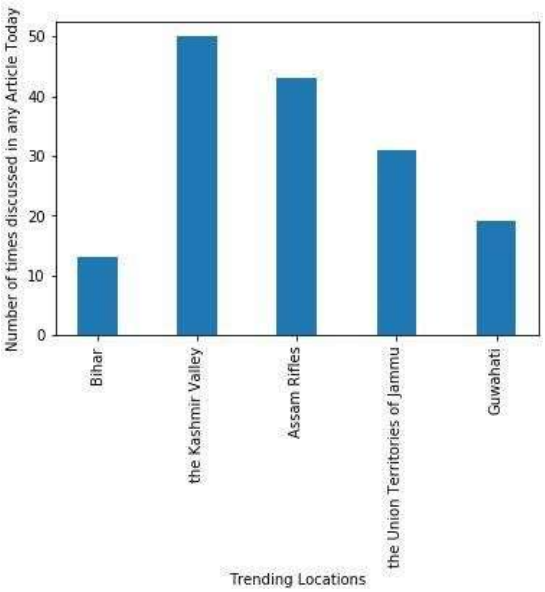

Figure 5: Timeframe 5 (CAA/CAB Protest)

The final timeframe was the one where the Indian Government introduced CAA (Citizenship Amendment Act) and was trying to implement NRC (National Register of Citizens). The bill had a huge protest related history in Assam and again CAA since it excluded Muslim refugee's citizenship saying citizenship is being offered only to minority religion from Pakistan, Afghanistan and Bangladesh. It created a huge uproar. The algorithm was successful in detecting Assam, Guwahati and Kashmir. The performance of the algorithm was decent in this timeframe.

1. RESULTS OF TRENDING KEYWORD DETECTION

The results of the above discussed Trending Keyword Detection Algorithm when used in all the timeframes are as follows-:

```
[('the Srinagar-Jammu highway', 61),
 ('the Indian Central Reserve Police Force', 60),
 ('Pakistani', 58),
 ('Pulwama district', 45),
 ('CRPF DG R.R. Bhatnagar', 40),
 ('Jammu & Kashmir', 32),
 ('U.S. State', 22),
 ('Rajnath Singh', 19),
 ('Jaish', 16),
 ('Arun Jaitley', 14),
 ('Narendra Modi', 13),
 ('Satya Pal Malik', 10),
 ('the Union Home Ministry', 10),
 ('Kulbhushan Jadhav', 9),
 ('"New Delhi', 8)]
```

Figure 6: Trending Keywords (Pulwama Attack)

The algorithm successfully managed to capture the Srinagar-Jammu Highway where the event took place, Pulwama district, CRPF (who were targeted) and many other useful keywords thus providing very good context of the event.

```
[('Pakistani', 18460),
 ('India]222Jayati', 14800),
 ('Indians', 12834),
 ('aircraft', 8424),
 ('Ministerial', 6784),
 ('government)."The', 5830),
 ('Campaign]508Sivakumar', 5778),
 ('iafstrike', 5184),
 ('PulwamaAttack', 5166),
 ('Enforcement', 4888),
 ('terrorism', 4532),
 ('Court]2A.', 4396),
 ('Chiefs', 3872),
 ('"Party', 3807),
 ('strikeresource', 3599),
 ('Friday', 3552),
 ('dissatisfaction', 3504),
 ('border', 3312),
 ('Delhi]10Abdul', 3306),
 ('official]329Nesar', 3285)]
```

Figure 7: Trending Keywords (Balakot Strike)

The algorithm again is successful in capturing some of the most important details of the event by capturing "iafairstrike", "PulwamaAttack", "terrorism", "aircraft", "dissatisfaction" and "border". Thus, hinting on border tension, air strikes and the main reason behind it all Pulwama Attack.

```
[('stateswoman', 5375),
 ('ModiKillingKashmiris', 4928),
 ('Minister', 4633),
 ('JammuAndKashmir', 3900),
 ('government', 3465),
 ('party-', 3444),
 ('Tuesday', 3358),
 ('Articles', 3330),
 ('Scindia', 2664),
 ('indian)."she', 2318),
 ('Union', 1827),
 ('J&K."We', 1800),
 ('leadership', 1775),
 ('bill', 1694),
 ('KashmirFalse', 1560),
 ('people', 1440),
 ('Presidential', 1431),
 ('Congress', 1403),
 ('courtroom', 1220),
 ('Sabha', 1128)]
```

Figure 8: Trending Keywords (Scrapping of Article 370)

The results again reflect some signs of protests, references to bill, Jammu & Kashmir. Although the results provide some

information, the performance of the algorithm in this timeframe is decent.

```
[('Indians', 15408),
 ('courtyard', 14504),
 ('government', 7239),
 ('ministership', 6858),
 ('party', 6837),
 ('Ayodhya', 5980),
 ('verdict', 5334),
 ('Ramjanmabhoomi', 4950),
 ('Supreme', 4935),
 ('temple', 4455),
 ('mischief', 3648),
 ('countryman', 3290),
 ('statement', 3240),
 ('Hindus"', 2916),
 ('case', 2772),
 ('year', 2700),
 ('Singhal', 2324),
 ('president', 2310),
 ('Thursday', 2280),
 ('Lawyers', 2278)]
```

Figure 9: Trending Keywords (Ram Mandir vs Babri Masjid Verdict)

Compared to last timeframe, the algorithm performs significantly better in providing context of the event. The algorithm is successful in catching "verdict", "courtyard" which was again disputed land, "Hindus", "Ayodhya", "Supreme" referencing Supreme Court, "temple", "Lawyers" etc. The algorithm also managed to capture the day of the verdict as this was a big News Headline at that time.

```
[('years.#CABBill', 35814),
 ('LokSabha', 12871),
 ('Indians', 12610),
 ('Citizenship', 11907),
 ('statement', 11618),
 ('government', 10430),
 ('Minister', 9447),
 ('party', 7810),
 ('Monday', 6336),
 ('Amendment', 5432),
 ('country', 5016),
 ('people?"He', 4800),
 ('Courts', 4416),
 ('Congress(M', 4141),
 ('India.@OfficialPU', 3774),
 ('December(With', 3483),
 ('Wednesday', 3196),
 ('"Pakistan', 2535),
 ('hometown', 2460),
 ('Bangladeshis', 2394)]
```

Figure 10: Trending Keywords (CAB)

In this timeframe too, the performance of the algorithm has been good. It has successfully captured the focal point of the event that is CAB (Citizenship Amendment Bill) as the most trending keyword itself. By detecting keywords like "LokSabha", "Citizenship", "Bangladeshis", "government" etc. it is successful in providing knowledge that the event was a political one. When you see the results, major points of the CAB/CAA are all captured.

Now after seeing the results of Trending Location Detection Algorithm and Trending Keyword Detection Algorithm, we now have some data on what the crucks of all theses events were. We have successfully extracted useful information about the characteristics of that defines these events. All these locations and keywords of each event are representative of a major event that happened in that particular timeframe.

It was due to this reason, filtering News Articles that are representative of the same event was assumed to be Event Detection. As in existing literature too, be it the results of a Graph based Method or be it a Topic detection-based method, in an Event was said to be detected if all the News articles related to the event are somehow grouped/linked by the end of the approach.

Since we have useful information about the what is being discussed in bulk during this timeframe, we can filter out News Articles that consists of the same combination of things that are trending. This leads us to the final step in our Algorithm "Filtering Related News Articles".

D. FILTERING RELATED NEWS ARTICLES

For filtering News Articles, there must be a value attached to each of the news articles and there must be a threshold that bars those articles who are for some reason unable to cross the barrier. For that purpose, we have defined the following metrics for each News Article.

$$relscore = 0.5 * keypercentage + 0.5 * locpercentage$$

Where

$$keypercentage = \frac{No.\ of\ matching/substring\ keywords}{Total\ number\ of\ keywords\ in\ the\ document}$$

$$locpercentage = \frac{No.\ of\ matching/substring\ locations}{Total\ number\ of\ locations\ in\ the\ document}$$

for evaluation of the correctness of web search results are chosen. Hence precision, recall and F-measure (calculated with the help of precision and recall) is chosen as the evaluation metric.

Although all the metrics are calculated but our main focus has been to detect the correct event rather than getting all relevant documents. Hence in the table below it can be observed that although in many cases the recall of the algorithm has been good but the consistency of good results is present only in Precision Column of the table.

For each timeframe, the dominance of the related articles in the timeframe is also show. Dominance is defined as the percentage of articles present in the timeframe that are actually related to the event. This is shown to depict the correctness of the algorithm and prove that the timeframes being used were not manipulated to get the desired results. The following table describes the results.

relscore is the match between the trending characteristics like location and keywords and those found in the News Article. In this work, it is assumed that both locations and keywords found in the articles have an equal contribution in defining the event hence relscore is calculated by taking an average of both. Key percentage is defined as the percentage of keywords match found between keywords found in News Articles and those found in Trending Keywords. Similarly, locpercentage is defined as the percentage of location match found between locations found in News Article and those in Trending Locations.

Each News Article is assigned relscore value and after some experimentation, a threshold of 0.4 was set and the News Articles from these timeframes were filtered. For each timeframe, a filtered timeframe was created which consisted of News Articles who crossed the threshold barrier.

V. EVALUATION METRIC

Since all the work done for the purpose of Event Detection was done using unannotated data i.e. News Articles, for evaluation purposes, each of these articles were manually annotated. For each timeframe, the News Articles were divided into two classes; Class 0 and Class 1 with Class 0 representing unrelated article and Class 1 representing related articles. All the above discussed algorithms were implemented and for evaluation purposes, filtered timeframes were used.

For evaluation purposes, we have chosen precision and recall as to be the metric on which the algorithms performance is evaluated. Based on these two parameters F- measure is calculated and is considered to be the ultimate performance metric. Since filtering relevant News Articles from a timeframe is very similar to filtering relevant documents of a web search result, the metrics that are used.

Table 1: Evaluation Results

Timeframe	Dominance	Precision	Recall	F-measure
Pulwama Attack	35.21%	78.26%	72%	75%
Balakot Strike	34.1%	71.8%	52.27%	60.52%
Scrapping of Article 370	44%	90%	81.81%	85.71%
Ram Mandir vs Babri Masjid verdict	36%	75%	51.16%	61.11%
CAA/CAB Protests	45%	65%	25%	36%

VI. CONCLUSION

In this paper, we propose a simplistic approach which combines the advantages provided by using a timeframe along with the results obtained from superstring matching algorithm on the key characteristics of any event i.e. location and keywords to detect the trending information in the timeframe. Along with providing efficient methods for detecting the trending characteristics of an Event, this paper also provides a novel filtering-based approach to filter out the timeframe and extract just the event related News Articles by using a novel metric "relatedness score (relscore)" which takes into consideration the characteristics of the article in order to get its relatedness to the trending characteristics of the timeframe. Thus, helping us to detect the occurrence of any radical/major event in a given timeframe.

VII. REFERENCES

1. Yifang Wei, Lisa Singh, Brian Gallagher and David Buttler, "Overlapping Target Event and Story Line Detection of Online Newspaper Articles". In IEEE International Conference on Data Science and Advanced Analytics (DSAA),2016, pages=222-232
2. J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in SIGIR. ACM, 1998, pp. 37–45.
3. Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and online event detection," in SIGIR. ACM, 1998, pp. 28–36.
4. T. Brants, F. Chen, and A. Farahat, "A system for new event detection," in SIGIR. ACM, 2003, pp. 330–337.
5. G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," in VLDB. VLDB Endowment, 2005, pp. 181–192.
6. T. Lappas, B. Arai, M. Platakis, D. Kotsakos, and D. Gunopulos, "On burstiness-aware search for document sequences," in KDD. ACM, 2009, pp. 477–486.
7. T. Lappas, M. R. Vieira, D. Gunopulos, and V. J. Tsotras, "On the spatiotemporal burstiness of terms," in VLDB, 2012, pp. 836–847.
8. J. Weng and B.-S. Lee, "Event detection in twitter." ICWSM, vol. 11, pp. 401–408, 2011.
9. T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in WWW. ACM, 2010, pp. 851–860.
10. N. Ramakrishnan, P. Butler, S. Muthiah, N. Self, R. Khandpur, P. Saraf, W. Wang, J. Cadena, A. Vullikanti, G. Korkmaz et al., "'beating the news' with embers: Forecasting civil unrest using open source indicators," in KDD. ACM, 2014, pp. 1799–1808.
11. J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme- tracking and the dynamics of the news cycle," in KDD. ACM, 2009, pp. 497–506.
12. C. Li, A. Sun, and A. Datta, "Twevent: segment-based event detection from tweets," in CIKM. ACM, 2012, pp. 155–164.

13. H. Sayyadi, M. Hurst, and A. Maykov, "Event detection and tracking in social streams." in ICWSM, 2009.
14. F. Xu, H. Uszkoreit, and H. Li, "Automatic event and relation detection with seeds of varying complexity," in AAAI workshop event extraction and synthesis, 2006, pp. 12–17.
15. Y. Nishihara, K. Sato, and W. Sunayama, "Event extraction and visualization for obtaining personal experiences from blogs," in Human Interface and the Management of Information. Information and Interaction. Springer, 2009, pp. 315–324.
16. A. Ritter, O. Etzioni, S. Clark et al., "Open domain event extraction from twitter," in KDD. ACM, 2012, pp. 1104–1112.
17. K. Lei, R. Khadiwala, and K.-C. T. Chang, "A twitter- based event detection and analysis system," in ICDE, 2012.
18. S. Muthiah, B. Huang, J. Arredondo, D. Mares, L. Getoor, G. Katz, and N. Ramakrishnan, "Planned protest modeling in news and social media." in AAAI, 2015, pp. 3920–3927.
19. D. Wang and W. Ding, "A hierarchical pattern learning framework for forecasting extreme weather events," in ICDM. IEEE, 2015, pp. 1021– 1026

.