International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 10 (2018) pp. 208-218
© MIR Labs, www.mirlabs.net/ijcisim/index.html

Submitted: 31 Jan, 2018; Accepted: 5 August, 2018; Publish: 10 August, 2018

# Semantic Filtering and Event Extraction of Twitter Streams through RDF and SPARQL

**N. SenthilKumar and Dinakaran M**
**School of Information Technology & Engineering**
**VIT University, Vellore, India**
**Email: senthilkumar.n@vit.ac.in, dinakaran.m@vit.ac.in**

***Abstract***

For long, there has been a huge demand to develop an efficient mechanism to effectively search and extract much needed information from the social web. Manual annotation is effectively possible in information retrieval for limited number of documents, but it is impractical for large accumulation of document retrieval particularly from social media. The principle objective of this research is to comprehend the emergency news by semantically extracting entities and its relations in the posted user generated content. In order to achieve this, the two most sought process of this enablement is that key event identification of every news items and semantic element extraction from those items. These two processes paves the way for building an effective knowledge base as well as a semantic retrieval engines to augment the event level semantic filtration of news items. It has been well observed and noted at many instances that social media's prevalence is a major source of collective formation of large public opinions. Nevertheless, deriving much needed information from it is very useful because of the fact that it is fresh and above all, no one has mediated its content. It is not the news that we have pondered for, but the views of millions of people on the particular events. Hence, this research has turned to the new dimension of identifying the events which are deemed important for many levels of processes. We discover the events by clustering similar information from mainstream social media and categorize the events semantically to enrich the whole process to obtain the much sought discovery of hidden information.

**Keywords:** ***RDF, SPARQL, Jena, Semantic Web, Tweets, NLP***

## I. Introduction

The social media's dominance has been proportionally increasing every day and its thumb of rule is deemed with high privileges in bring the decision making process to the core development. There were many impeding problems in our conventional news streams as reported in the paper [2] that the news items had been scattered in too many places and invariably repeated at many times. We could not get the real facts of the events but we should sit on the shoulder of the reporters to view the details. But in the case of social media, everyone has their own view of the incident or event and immediately posted on the social media. We can get the diverse opinions of the same incident or event and it has been very evident that the posts are the views of the users but not the news. It has also been statistically analyzed by researchers [8] [12] that the reported news and user generated content are stand far apart in bringing the realistic view of the incidents. Social media has covered the wide range of discussion of the incidents and many eye-witnesses had been shared their thoughts and opinions for the event.

So, in our research, we have taken the twitter as the social media platform and extract the tweets for the events collectively. To enable the extraction process effectively, we continuously crawl for the information of the events and further clean and store the content in the data store. Besides, we also look for sub-events that are comes under the topics of the event [13] and collect the related sub-events to augment the decision making process substantial. In doing all this accumulations, the geographic and temporal relations of the specified events would be obtained to set the precise estimation of occurrences. But many of the times, it has been noted [4] [6] that the redundant information from the streams are very high and it has taken elapsed execution time and demands unusual memory space which is most of the times obstruct the automated computation of the process [19]. Above all, the major task of this research would be focused on extracting the potential named entities from the Twitter Streams and effectively disambiguate the candidate mentions occurred in the knowledge sources like DBpedia, YAGO, etc. We have applied the classification algorithms [3] and similarity measure techniques [20] [21] to easily relate the extracted entities from tweets with DBpedia candidate mentions. Eventually, we have illustrated the semantic enrichment of tweets and how it has been the driving force for semantic search of potential information.

## II. Related Works

According to [4], identifying the events in the continuous user generated streams gives the serious difficulties and posed several problems to identify the sub events from news streams. Identifying the events from the largely posted user's content has implicitly concerns about discovering the unidentified events and it is named as retrospective identification. The other category of identifying the events is through live feeds from social media content called On-line identification. Event identification has been challenge for many information retrieval problems as mentioned in [7] [11] and it gives the range of difficulties like integrating the information from more one source, collecting the details of the content such as spatial and temporal values, identifying the sub-events for the incident and classifies the event to other category of events.

In the paper [15], it has been well noted that event identification uses the statistical methods to observe the events history and conclude on some proximity based estimation of preciseness. According to the authors, they had proposed a robust system that employs an algorithmic method Latent Dirichlet Allocation which monitor the events and omit the content that is not relevant to the event. In their proposed algorithm [15], they compute the average Euclidean distance between the events and segregate the abnormal changes present in the streams so that unknown distribution of data would be neglected and get the Bag-of-Words. Another approach that they had proposed in the algorithm is the log-likelihood rate by which the statistical ratio of data and TF-IDF difference present in the user generated streams can be normalized based on term weighting and similarity score.

To the latest [11], the classification algorithms have been used largely for the event identification and classification. Many supervised leaning algorithms had been applied to divide the user generated streams into the already defined topic categories and carry out the detection process much easier than before. The strategic approaches such as text classification and named entity recognitions have been extensively employed for exploring the hidden events and disambiguate the selection process (uses vector space model to increase the viability of event detection).

The recent proposal [33] suggest that the effective way to access the RDF data is by storing it on the Cloud based platforms and obtained the results using Hive, Scala, Spark-SQL and Impala. They had conducted an extensive review of the relational models of the RDF storage and productively carried out to address the semantic filtering challenges. The authors in [33] also designed the framework to convert the database into the RDF using a system called DB2RDF that is specific to entity oriented schema for RDF data. However, the performance of the semantic querying was not fulfilled and then proposed a new model called as S2RDF which is based on SPARQL. The semantic filtering of the results was performed on the basis of SPARQL query which is the combination of semantic permutations of properties (i.e., Subject-Subject, Subject-Object, Object-Subject and Object-Object).

## III. Proposed Methodology

The proposed model of our system is going to enable the decision making process more accurate and enable the semantic extraction of information from Twitter much simpler than before. The idea of the research is to extract the information from the twitter into the potential named entities and the system is able to automate the whole process to make the meaningful conversion of the events by semantic technologies. The proposed method of our work is given below:

***Algorithm - Entity Disambiguation and RDF Triplet Conversion***

**Input:** *Load and Store the tweets Tn of the Event E for the specified time limit*

**Output:** *Disambiguate the entities and generate the RDF triplet of the tweets*

```
Step 1: For each tweet $T_i$ from the Collection $T_n$
Step 2: Identify the potential named entities and classify
        them using ANNIE.
Step 3: Identify the core details of the events such as event,
        event type, location, resources, time, region, etc.
Step 4: Find the relatedness score of each entity extracted
        using DBpedia URI.
Step 5: Find the near-proximity score of each entity with
        possible candidate mentions listed for projection by
        DBpedia source.
Step 6: Compute the similarity score between entity and
        mention.
Step 7: Convert the entities of the tweets to the appropriate
        RDF Trifle conversion.
Step 8: Store the generated RDF Triple for the tweets into a
        RDF Store (Jena/ Sesame).
Step 9: Decision can be obtained through the SPARQL
        Query Application
```

Here, the semantic web technology [7] helps to bring the potential meaning of the information posted on the social media. The RDF Triple format and SPARQL query [1] would be the key aspects of our research method since both paves the ultimate design of the approach. The RDF Triple [17] [27] can make the sentence meaningful by converting into equivalent graph which could turn the machine understand easier than humans and this would help for automation of whole process execution and get the precise answers from SPARQL query.

### *A. Problem Definition*

We define our problem as follows:

> Given an event $_e$ of any situation $_S$, extracts the tweets $_T$ from twitter streams for $_K$ number of users and convert the tweets $_T$ into the semantically meaningful conversion called $_R$ where $_R$ denotes the RDF conversion of the tweets that extracted. Each tweets $_T$ has been mentioning different aspects of the event $_e$ since $_K$ is increasing proportionately.

The sole objective of the research is to give the meaning of every post that the user has been generated in the social media and shun the ambiguity pertaining over the user generated content. The major focus of the research would be on removing the ambiguity in the twitter streams and that would make the whole process effective in terms of the real objectivity of the events.

### *B. Pertaining Challenges*

Since, the extracted tweets from twitter has been very open and easy but it encounters huge amount of challenges to overcome. In the paper [7] [32], the authors had pointed out precisely that there would be no metrics stipulated about how much of information need to monitored, extracted and evaluated. The following are the major problems that give various challenges in yielding the results:

a) Extracting tweets covers different aspects of the event and failed to measure the distinction between the choices of the events.
b) Many tweets are tending to be very noisy and sometimes very irrelevant to the event and which causes unnecessary computation problems.
c) No measure to counteract rumors germinated during the events and it has been feared that it would spread vehemently over the short course of period.
d) Dealing with misspelled tweets is a big task since there had been no apt use of dictionary to make over.

Hence, the proposed model is covering the perpetual challenges lasted for long and observed many serious outcomes of events which has been very critical for evaluation.

### *C. Proposed Model*

We have taken Twitter as our social media platform and continuously crawl for the tweets that are related to the events. Accumulating tweets can divide the process into two major categories (i.e. Extraction Pipeline). One, the proposed system is going to look for the tweets related to the events and store in the RDF Store called Jena [1]. Second, need to eliminate the tweets that are duplicated and identify the tweets that are not all related to the events. Here, the distinction is required to understand the two approaches. The tweet duplication can be very easily detected with similarity index as stated in [21] and that can be ousted out but identifying the events that are not related to events is the tough tasks. We have approached this context with algorithmic approach given in [13] [28] that deals with taking the geographic and temporal relations of the events. In addition to that, the context in which it has been taken and relationship to the posted events are counted for connection before judging the irrelevancy of the tweets.

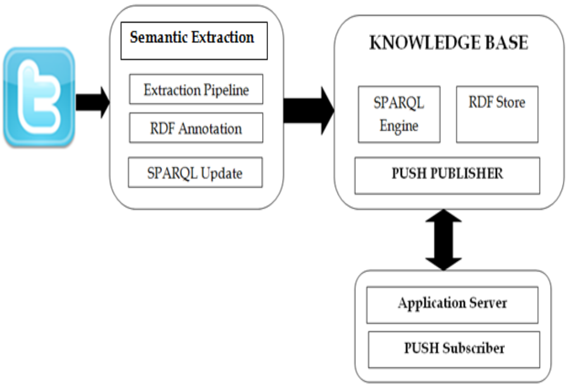

***Figure 1: A Proposed Model for Semantic Understanding of Twitter Streams***

Now, applying the semantic technologies [7] to increase the potential comprehension of the stored tweets for the event and generate the RDF triple for every tweet that have been extracted (i.e., filtered tweets after the two step process of categorization). The RDF Triple would generate the RDF Graph for every tweet and gives the best knowledge of the events by traversing the path between entities which have been fixed as named entities of the events. According to paper [1], the RDF entailment of the tweet can enrich the information and make the system understand the whole event in the meaningful context and yield the results with absolute precision. To get the meaningful information, we need to supply the SPARQL Query to the RDF Store [17], which in turn pushes the results. Here again, we need not give the SPARQL Query to the system since many user have no knowledge of the SPARQL query. Hence, we proposed an application which would take your question and generate the equivalent SPARQL Query and yield the potentially meaningful answer (i.e., answer with no ambiguity). The proposed work aims to shun ambiguity pertaining in the event and supply the results with preciseness. Ambiguity poses many challenges in information retrieval and we have overcome this difficulty by implementing semantic based retrieval (RDF Triple and SPARQL takes this process to get rid of ambiguity and make the content potentially meaningful).

## IV. Detailed Description

In this section, the four components of the proposed research, Pre-processing effects, Event Extraction, Named Entity Recognition and RDF Triple conversion for tweets has been explained clearly with examples.

### *A. Pre-processing Effects*

Unlike the pre-processing of natural language text, tweets pre-processing is the most complex and complicated task ever before. In the conventional natural language text [16], we can identify the sentence by using the WordNet and check the sense of every word present in the sentence. The ambiguity prevalence in natural text is very high and that posed many challenges to the researchers [17] [22] to retrieve the meanings out of it. But in the case of tweet pre-processing, it has been totally naïve and complicated in terms of obtaining the word sense of the tweet. The tweet consists of canonical terms and words which not possible to cross check against any dictionary. Besides, the grammatical context is largely missed in twitter streams and posed an upheaval conundrum to the analyst to get through the actual meaning of it. Hence we have taken the following measures to get the work and pre-process the tweets effectively.

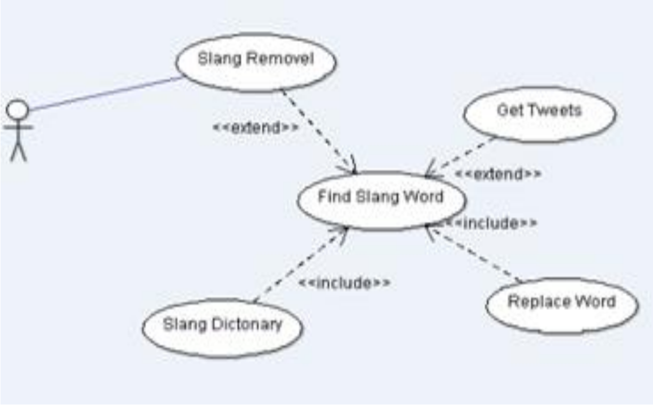

***Figure 2: Preprocessing and Normalizing the Tweets***

Using Twitter API, we are going to fetch the RAW tweets from Twitter. Tweets are collected from twitter which is publicly available in [6]. When the tweets are collected and stored, then the very next process tends about data cleaning or preprocessing, [3] where it deals about the normalizing the tweets so that it will be easy for identifying the potential named entities out of tweets and hence we have done the pre-processing task very strenuously (i.e., removal of punctuations, url links, some symbols and canonical terms). The slang words are the biggest challenge in extracting the facts out of tweets and dealing with the slang words is a yet another challenge which we faced in this project. The slang words are very short and very canonical in term of its presence and it needs to be appropriately replaced with corresponding words in the slang dictionary. In this regard, we have constructed the slang dictionary for all the slangs in the online social forum and used it for the pre-processing steps.

The principle task of the project lies in identifying the potential named entities from tweets and make them use for the effective classification of tweets for sentiment analysis. So we have collected the named entities from the preprocessed and cleaned data sets, and used Alchemy API key which is used to get the entity collections. There were many chances that the ambiguous words prevalently present in the tweets and the task of disambiguating the words into the proper surface of the word is the seminal work of the project [13]. The disambiguation is the process of clear the data content tweets with the help of DBpedia via using its references which is based on some Tag remover algorithm.

For every tweet, we do the following to pre-process it:

a) Remove the hyperlinks(e.g., http, https, ftp, etc) and take only atomic elements

b) Check the correct spellings for the canonical terms with the dictionary which developed for twitter streams.(E.g., luv -> love, cum – come, etc) and convert into original word sense(In this case, we had used noslang and urbandictionary website for informal words lookup).

c) Eliminates symbols and stop words because it has not all related to our proposed work. Stop words would be the hindrance for finding the named entity recognition.

d) Apply POS tagger to split the tweet and identify the named entities present in the tweet.

### *B. Event Extraction*

The event extraction from Twitter is carried out through the Twitter Streaming API which is the standard application fulfills the filtration process effectively [6] [30]. It extracts the tweets for the event by crawling through the hashtags created for the events by various NEWS sources in twitter and monitors the user generated posts subsequently [9]. The crawler is started to fetch the tweets that it has been monitored and stored in the data store. For each and every tweet that it has fetched would consists of the original post, author, timestamp, geographic information and hashtag from which it has been obtained. All these properties are very useful in deriving the patterns and identify the purpose of the posts.

The formal definition for entity extraction of twitter streams is expressed in the graph theory as G(E,V) where E and V represents some set of edges for the given set of vertices. To determine the potential social entities prevailed in the twitter streams and build the appropriate relationships between the entities, we link the set of edges E, in which $v_i \in V$ i = 1. . . N denotes the extracted entities in twitter streams and $v_iv_j \in E$ denotes relationship between entities $v_i$, $v_j \in V$. In this connection, to estimate the candidate entity for the query q, the search engines would normally generates most ambiguous entity sets about the given candidate entity and it is termed as

$$a_i = q \leftarrow a_i \qquad (1)$$

However, in our proposed approach, we have introduced the novel method to tackle this ambiguity prevailed over the search results by incurring the semantic web ontology for the domain at which the entity has been dealt with through the appropriate level of ontological weight and it can drastically reduce with the addition of ontology candidate keyword kw, i.e. consequent of

$$aw_i = q \leftarrow a_i, KW \qquad (2)$$

Which reduced the entity ambiguity with which $|aw_i| \leq |a_i|$, $|a_i| \in a_i$ is a cardinality of $a_i$ and $|aw_i| \in aw_i$ is a cardinality of $a_i$, kw . By utilizing the well-formed query for the candidate entity in the query, the named entity information would come as

$$a_{"i"} = q \leftarrow "a_i" \qquad (3)$$

In some cases, $|a_{"i"}| \leq |a_j|$ $and$ $|a_{"i"}| \in a_{"i"}$ is a cardinalty of “$a_i$” [14]. As well as with

$$aw_{"i"} = q \leftarrow "\mathrm{a_i}", KW \qquad (4)$$

is about one of information concentrations of a named entity. Then after the pruning of the entity cardinality, in the next process, the relationship between two named entities is based on the concept of co-occurrence. Thus,

$$a_i \; a_j = q \; \leftarrow a_i, a_j \qquad (5)$$

Which is a process to augment the semantic similarity between the two named entities and build the relationships between them, with which $\left|a_i \cap \; a_j\right| \leq |\; a_i| and \left|\; a_i \; \cap \; a_j \;\right| \leq |a_j|$ and$\left|a_i \cap \; a_j\right| \in \; a_i a_j$ is a cardinality of $a_i a_j$. Besides, with the supplement of a keyword towards the co-occurrence will usually subsides the number of entities given, and that is

$$aw_i aw_j = q \; \leftarrow q a_i, KW \qquad (6)$$

But it should satisfy that $\left|aw_i \cap aw_j\right| \leq \left|a_i \cap \; a_j\right|, \left|aw_i \cap aw_j\right| \in aw_i aw_j$ is a cardinality of $a_i a_j$, KW [6]. Similarly, the effective utilization of the well-defined entity set for the query will yiled the appropriate relationships between the two named entities correspondingly.

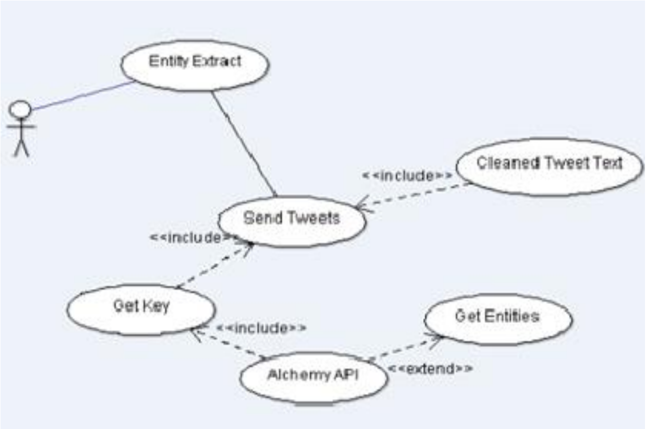

***Figure 3: Detect the entities from tweets using the NER Tools***

### *C. Named Entity Recognition*

The named entity recognition has been the backbone of the proposed work since it is going to identify the named entities present in the tweet. There are very high chances and prevalence of the ambiguities in determining the named entities chosen from the context. The named entities can fit to the domains like person, country, organization, date, location, percentage, money and any real world generic object as stated in [11] [29]. Hence, choosing the named entities would determine the scope of the proposed work. To compute the overall matching between the two distinct topics, we calculate the semantic similarity (SemSim) between the two topics $\mathrm{T_i}$ and $\mathrm{T_j}$, by dividing the sum of the similarity score of the candidate sets. In order to be very careful, we have taken the following measure in implementing this out.

$$SemSim = \frac{2*\mathrm{Compare(Ti,Tj)}}{|\mathrm{Ti}| + |\mathrm{Tj}|} \qquad (7)$$

The DBpedia Ontology would be very useful to take the correct choice of picking the named entities from the context. The DBpedia Ontology has more than million entities registered and manipulated internally in its construct. This paves the way for taking the correct named entities for our event and classifies our tweets into this predefined categorization of named entities. So, the task underlying in this part is that it would split the tweet into token using POS Tagger( i.e., the POS Tagger would split the sentences into tokens and assign the subject, verb and noun to the tokens) and for every subject and noun, the system probe for equivalent named entities in DBpedia Ontology.

***Example:***

The Prime Minister Narendra Modi attended the Economic Forum in Geneva on 7th Nov 2016

⇩

<namefind/person>Narendra Modi</namefind/person>
<namefind/location>Economic Forum</namefind/location>
<namefind/location>Geneva</namefind/location>
<namefind/date>7th November 2016</namefind/date>

### *D. The Proposed Approach for Entity Detection*

The seminal task of detecting the appropriate named entities from the collection of tweets is prevalently challenging and classifying them into its domain would yield a upheaval task to the system. Hence, in our proposed work, we have carried a task of eliminating the risk underlined in the proposed approaches given in [3] [9] [22] and brisk the focus on emanating the lightweight POS tagging approaches to effectively identify the named entities from the tweets. In order to streamline the process, we have applied the Naïve Bayes Classifier [22] to extract the entities from the tweets and classify them as potential named entities. To substantiate about its potential scope of the entities, we have enabled an approach which identifies the possible links in the knowledge source like DBpedia. The Naïve Bayes Classifier can extract the potential named entities from the tweets considering the following approaches:

a) Check the occurrence of the entities extracted from tweets on the WordNet.
b) Choose the most relevant DBpedia URI for the given entity.
c) If the named entities is not present both in WordNet and DBpedia, look for the valid link in any website for reference and fetch the summary of the same to augment the entity presence.

The Named Entity Recognition tools have given us the option to explore the results and interpret the means of operations carried on such domains but they failed to give the accurate results in terms of precision and recall. Hence, we have taken this approach to enhance the standards of the approaches and yield the good results. The below table 1 shows us the accuracy rate of the NER techniques followed in matching the named entities with existing knowledge sources like DBpedia, YAGO, etc.

***Table 1: Performance of Entity Detection Techniques***

| *Entity Detection Technique* | *Accuracy* |
|---|---|
| ARK POS TAGGER | 77% |
| T-NER POS TAGGER | 92% |
| ARK + T-NER (Merged) | 98% |

In our proposed approach, we have taken the candidate entity sets and determine the function to filter the potential entities from tweets based on the context and the relevance of the events. Moreover, the accuracy of the named entity selection would be considered after applying the entity selection function that would set either related or not related.

$$f(t) = T \rightarrow \{related|\, not\ related\}$$

Therefore, we have used the ARK POS Tagger coupled with T-NER POS Tagger to filter the entities from tweets and map the extracted entities into predefined categories. The relatedness score of the entity is very crucial so as to deduce the ambiguity persist over on the entities.

### *E. Feature Extractor & Entity Linking*

Once the relatedness score and the entity similarity score of the entity is determined, we now need to bridge the link between the entity and DBpedia mention. But, link the entity with DBpedia [22] [34] would give us the ambiguous results and paves ways for wrong connection of mentions. In our proposed approach, we have taken the context and entity relatedness together for matching against the DBpedia source and detect the near proximity of mention for the given entity. In order to map the entity and mention, it had already been noted that for many entities, the DBpedia has holding many similar mentions of the same name ( 1- to – M cardinality) and also the entity would also have many candidate meanings and links to different DBpedia URIs. For instance, “Jaguar” can be linked to ‘Animal’ or ‘Car’ or mapped with a pop song named Jaguar.

To annotate the tweets, we have found that most of the existing systems had used the external sources such as Knowledge Bases or Dictionaries to appropriately disambiguate the extracted entities and in many cases, it has failed drastically due to the NIL match of the those mentions in the used knowledge base. Hence, we proposed a new model by which we can resolve this difficulty and increase the matching proximity to greater extent.

The idea is that whenever we want to match the extracted entities from tweets into the knowledge base mentions, we have low accuracy and high ambiguity. Hence, first we cluster all the tweets and understand that the whole tweets topic and subtopic and when the event was taken place and what are the arguments places on those tweets and location and objects involved. Based on these factors, we fetch the news articles and store it in the database. Then we split the news articles by each sentence and identify the named entity in each sentence to RDF file. Store the complete RDF file for the collected news articles and create the ontology class for the same. Once this is ready, we can very easily match the NIL results to the appropriate entities and increased the accuracy.

This ontology is now become act as knowledge source for our disambiguation. When we process each and every tweet, we find the exact match of those entities against the knowledge source such as DBpedia or YAGO. If it is not present, then it sends the NIL results. Now by means of our proposed method, we can again cross match with our own ontology created from news articles and find the exact match of those entities. In this method, the accuracy is relatively high because the created ontology is extracted from news articles related to the tweets and context of the news articles is high relevant and appropriated match with the tweets. If we go for entity – mention match with DBpedia, it has list out candidate mentions for the entity and we need to probe for the context pertaining to the tweet. But if we match the same with our own ontology, it is exact and gives appropriate match.

Hence, in our approach we have taken the link probability [17] [31] for the entity with DBpedia mention and it can be defined as follows:

$$F_{(e,m)} = \frac{Count(m,e)}{Count(m)} \qquad (8)$$

Here, we utilized an outlined ontology to arrange the mentions for the given named entities and appropriately estimate the similarity distance between them. Now the task is to estimate the distance between the entity and the suggested set of mentions from DBpedia. In this connection, we have taken the Cosine Similarity measure to access the similarity difference exists between the entity and candidate mentions.

$$CosSim(e,m) = \frac{Product(e,m)}{||e||*||m||} \qquad (9)$$

By this method, we have categorically filter the exact match of mention for the given entity and appropriately referenced with DBpedia URI as stated in [22]. We have utilized the DBpedia Spotlight to get the URI match of each entity and return the JSON results for our implementation.

```
def filter(entity):
        return JSON
(DBpediaSpotlight.annotate(entity));
```

The result of the proposed approach would create a binary mapping of the entity and mentions as seen in below table 2.

***Table 2: Identifying the relation between named entity and candidate mention***

| *Mention* | *NE Class* | *NE Link* | *DBpedia Ontology Class* | *Score* |
|---|---|---|---|---|
| Barack Obama | Person | Dbpedia: Obama, USA | Dbpedia-owl: Person | 3 |

| | | | | |
|---|---|---|---|---|
| Chennai | Location | Dbpedia: Chennai, India | Dbpedia-owl: Place | 1 |
| Cricket | Sports | Dbpedia: Cricket | Dbpedia-owl: Sports | 2 |

Generally, entities in DBpedia have its name, label, type etc and to fetch the entity name given in the DBpedia for the specified URI, it can be queried through the SPAQRL query as:

```
Select distinct *
where {
        ?URI rdf:label ?name
        ?URI dbpprop:iupacname ?name
        filter(str(?name) = “Sachin Tendulkar”)
        }
```

In order to get the category of the given entity from the DBpedia, we can give the SPARQL query as:

```
Select *
where{
        <http://dbpedia.org/resource/Vehicle>
        <http://purl.org/dc/terms/subject>
        ?categories.
        }
```

## V. Semantic Filtering

The semantic filtering task can be break down into two components. First, the semantic filter name given in the SPARQL query is used to represent the join pattern of the stored RDF results. Second components deals with correlative association between the fetched row for the name and its value where the row indicates the partition of RDF data and values matches the join pattern given in the query name. The semantic filter name given in SPARQL query delineated the association between properties and objects of the stored data. For example, the semantic filter name given in above SPARQL query is “Sachin Tendulker” which represent the join pattern of the query exactly matches with column name in the RDF data S. The SPARQL join operation is happen between the subjects, predicates and objects of the triples and the possible join correlation of triples would be Subject-Subject, Subject-Object, Object-Subject and Object –Object. Consider the following example which simplifies the execution of SPARQL query and its join operation.

```
        SELECT ? x ?y ?a ?b where
        {
        ?x      mention: Sachin_Tendulkar.
        ?x      tweet   ?y.
        ?y      :latitude  ?a.
        ?y      :longitutude  ?b.
        }
```

The above SPARQL query is simplified as follows:

x={mention_S, tweet_S), y={tweet_O, latitude_S, longitude_S}
where S and O indicated Subject and Object of the RDF triples and the semantic filter name “Sachin_Tendulkar” carried out all the possible join permutation on the properties tweet_O, latititude_S and longitude_S. The maximum number of semantic filter for the join operation is given below:

$$f(n) = n\,X\,(2^{n-1} - 1) \qquad (10)$$

### A. RDF Triple Conversion from Tweets

The final component of our proposed work is RDF triple generation [1] [7] for the extracted tweets. The RDF triples can makes the information meaningful and create a logical path for every entities identified. In our research, we have taken the tweets and generate the equivalent RDF Triplet so that we can have a hierarchical distribution of path for every identified named entity and produce the ontological construct to the entire tweets for the event. This would felicitate the decision making process meaningful and governs the fundamental setup of the information retrieval.

The proposed syntax for triple-like statements inside Twitter messages (“trippletweets”) is as follows:

```
Triple tweet := { subject |predicate | object}
subject := { @userid | #hashtag | http_uri }
predicate := { = | sameas | subtag |property |prefix }
object := { @userid | #hashtag | http_uri | "value" |prefix }
userid := [-_a-zA-Z0-9\.]+
hashtag := [-_a-zA-Z0-9\.]+
http_uri := http://[-_a-zA-Z0-9\./?&%#]+
property := [-_a-zA-Z0-9]+
prefix := { foaf: | tag: | gr: | sioc: | rdfs: | rdf: | skos: | owl: |
dc: }
value := "[^"]+"
```

**Example:**

Now, let’s take a sample tweet from Twitter and construct the RDF Triplet by considering the proposed syntax for Twitter.

***“No electricity in our city for last 2 days due 2 hurricane”***

For the above tweet, we need to identify the named entities and construct the RDF triplet for the same. Before we identify the named entities, we should pre-process the tweet using POS Tagger [13] which will split the tweets into atomic tokens by removing all the stop words, stemming the words and lemmatization. So it will be as follows:

***No Electricity, Our City, Two Days, Hurricane***

In this case, we can find some of the worst case for our analysis, i.e., our city and two days are sheer meaningless tokens and how the proposed system can take it meaningful. As we mentioned in the first component of the proposed work, we not only extract the tweets but also the properties of tweets like, location, timestamp and other geographical mention of the tweet. That would help here to convert the anonymous tokens in valid tokens. Hence, imagine that the city that mentioned in the tweet is New Delhi and the date of extraction of the tweet is 7th May 2015.

Now, the RDF Triplet for the given tweets is as follows:

| | | |
|---|---|---|
| T101: | hasResource: | No Electricity |
| T101: | hasLocation: | New Delhi |
| T101: | hasPeriod: | 7th May 2015 & 8th May 2015 |
| T101: | hasObject: | Hurricane |

Where, T101 denotes the particular one tweet and it would be unique for every tweet and that will bring the meaningful representation of entities present in the tweet.

Next step to the conversion of the RDF triple [7] to the tweets is Identifying the Triplet. The Triplet is made up of three properties: Subject, Predicate and Object (SPO). The subject denotes the tweet id, the object represent the named entities identified from the tweet and the predicate is the establishment of link between the subject and object. Here, the link would make a connection to the named entities and therefore more than one subject would link to the same named entities since we have been collecting the tweets for the event. So this will derive the hidden pattern and pop out the factual details to the user if asked. The RDF Triple can be represented in the tabular form [3] as follows:

***Table 3: RDF Triple Generation for the collected tweet***

| Subject | Predicate | Object |
|---|---|---|
| T101 | hasResource | No Electricity |
| T101 | hasLocation | New Delhi |
| T101 | hasPeriod | 7 May 2015 |
| T101 | hasPeriod | 8 May 2015 |
| T101 | hasObject | Hurricane |

Before to code the RDF Triple, there are certain ontologies required to include in the conversion process. The ontology requirements for this process are SIOC, Dublin Core, FOAF and many more as given in [7]. These ontologies are very useful in choosing the correct named entities and the corresponding relationship to the events. This would drastically eradicate the ambiguity problems pertaining in the twitter streams. The ontology given above possesses its own specifications and usage in the context of the event selection. The RDF Triple conversion process for the tweet would be written as follows:

```
<?xml version='1.0'?>
<rdf:RDF
xmlns:dc='http://purl.org/metadata/dublin_core#'
        xmlns:rdf='http://www.w3.org/1999/02/22-rdf-
        syntax-ns#‘
        xmlns: foaf=“http://xmlns.com/foaf/0.1/’
xmlns: sioc=“http://rdfs.org/sioc.ns/’
xmlns:h='http://xmlns.rdfinference.org/ril/Hurricane
#' >
<rdf:Description rdf:id='T101'>
        <sioc:resource> No Electricity
        </sioc:resource>
        <dc:location> New Delhi </sioc:location>
        <dc:period> Two Days</dc:period>
        <h:hname> Hurricane </h:hname>
        <dc:creator> Charles </dc:creator>
        <dc:date> 2014-22-12 </dc:date>
</rdf:Description>
</rdf:RDF>
```

The above RDF code would generate the graph like structure and represent the logical path for hierarchical execution of the system. The RDF code makes the machine understand the events on its own and automate the whole process independently. Hence, the proposed system has taken the implementation for the successful conversion of twitter streams into RDF triples. This novel approach would enable the decision making process effective and understand the events semantically better than before.

## B. *Empirical Test and Analysis*

We used Twitter4J API to gather disaster related tweets from Twitter and utilized TextRazor API to effectively recognize the potential named entities present over the tweets and link them accordingly to its respective DBpedia URI. Additionally, we used Stanford Core NLP Library to segregate tweet patterns with its rich natural language processing tools and performed sentiment analysis for grasping the sense of the tweets. Tweets were collected on the month of August 2017 and to witness the trust, we followed the leading news agencies on Twitter such as BBC World, CNN, New York Times, NDTV, and Breaking News. Tweets were crawled and stored only if it had at least one named entity that has its link on DBpedia URI. In our datasets, we were able to filter out 20 different topics and successively we classified the tweets based on seismic risk by applying the classification rules. The algorithm proposed above is able to detect the factual information containing about 3 out of 5 tweets.

***Table 4: Event relevance and categories***

| Event Category | Total Events | Potential Sub-Events by Relevance | | |
|---|---|---|---|---|
| | | R3 | R3+R2 | R3-R1 |
| Earthquake | 75 | 35(46%) | 51(68%) | 59(78%) |
| Tsunami | 120 | 46(38%) | 79(65%) | 88(73%) |
| Cyberattack | 114 | 51(44%) | 87(76%) | 95(83%) |
| Unrest in a Country | 150 | 77(51%) | 90(60%) | 97(64%) |
| Celebrity Death | 115 | 43(37%) | 66(57%) | 81(70%) |

| | | | | |
|---|---|---|---|---|
| Terror Attack | 120 | 68(56%) | 79(65%) | 85(70%) |

We tested the DBpedia corpus to identify potential events on seismic risk which provide the six complex event categories listed in Table 10. The entities were extracted based on the recommendations stated above and identified their relationship types in corresponding DBpedia URI. Besides, we again queried the DBpedia Knowledge Source for the sub-events correlated with the events extracted from the tweets. We substantially ranked the sub-events on the basis of frequency of occurrence and chose the best matched event category to the tweet. After evaluating the event categories against DBpedia, we determined whether the event is of positive instance or not. Sometimes, the retrieved events from tweets would pose a challenging task such as if it was partially relevant but not exactly appropriate to the categorized concepts. During these anomalies, we assigned the following three relevance scores in order to fit the events into their appropriate decks:

- Relevance (R1): Events with fuzzy relationship to the concept/category.
- Relevance (R2): Events with positive occurrences of sub-events or subject-object mapping.
- Relevance (R3): Events are positive instances and fit into the category for the posted query.
- Otherwise, the relevance zero indicates the events with absolutely NIL relationship.

Table 4 displays detected event categories and potential sub-events or co-occurrence of events with relevance score. As it was witnessed, the precision values varied considerably among the categories. The Stanford NLP Library deemed fit to extract the potentially relevant tweets and type filtering of events was absolutely effective at identifying the appropriate named entities. We obtained the accuracy of 74.13% and computed the Precision (0.641), Recall (0.716) and F-Measure (0.691) respectively for the given datasets.

### C. *Key Findings of the Research*

The dynamic change in the amount of information gathered at various medium of platforms indicates the need for rapid decision making process of crisis events. It has been observed that the information gotten from these sources rapidly varied every day. Statistics (Wirtz et al., 2014) has shown that the frequency of report variation grows ten times greater than the previous day. Besides, to better account for the report variation of the information accumulation, the report dimensions have been categorized into three crucial breakpoints, i.e., D + 1, D + 5, D + 10). This elapsed gap will fetch the detailed overview of the crisis or disaster based events and shown us the real potential of the event happenings (see Figure 3).

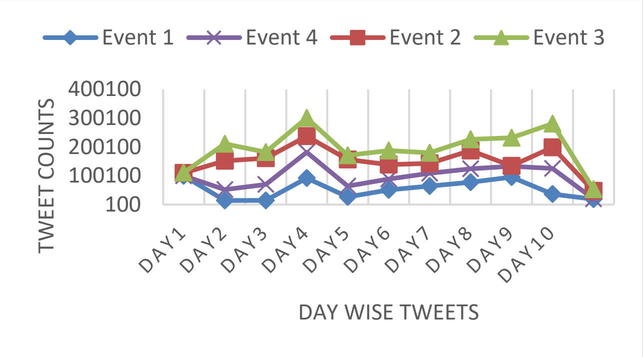

***Figure 3: Daily frequency of information on social media platforms***

Through the data which is obtained from various sources and on different days of report gathering, we can formulate a deviance of patterns and get through the details of anomalies that exist in the report. By applying the pruning algorithm, we can sort the crisis events for decision making process and get to the core base of the events. In this research, the real task is to find the actual reason for the crisis event and get the substantiated evidence for the occurring. To augment this process, we classified the events into many chronological orders and influenced the use of ontological background with semantic technologies. By mapping the different day event reports, we scrutinize the process for discrimination (i.e., fetch the positive or negative or neutral feedback from the potential users on the social media) and allow filtering the facts based on cross checking in tabulating the actual events of the situation.

## VI. Conclusion

The proposed work delineates the impact of semantic web technologies in social media since it has been widely acknowledge that the modern generation uses social media site like Twitter for long discussions and to produce effective decision making efforts. The sole objective of the research is to remove the ambiguity pertaining over the events and give the stake holder the corresponding information which would lead to further level of investigation or decisions. The two major focus of the research lies under named entity disambiguation and elimination of duplications in the twitter streams of the events. The proposed work has been carefully rendered in the direction of yielding the potential meaning of the events through RDF Triplets and obtaining the answers to the skeptical questions through SPARQL Query.

## References

[1] Elbassuoni, S., Ramanath, M., Schenkel, R., & Weikum, G. (2010). Searching RDF Graphs with SPARQL and Keywords. *IEEE Data Eng. Bull.*, *33*(1), 16-24.

[2] Cataldi, M., Di Caro, L., & Schifanella, C. (2010, July). Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the tenth*

*international workshop on multimedia data mining* (p. 4). ACM.

[3] Zenz, G., Zhou, X., Minack, E., Siberski, W., & Nejdl, W. (2009). From keywords to semantic queries—Incremental query construction on the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, *7*(3), 166-176.

[4] Allan, J., Carbonell, J. G., Doddington, G., Yamron, J., & Yang, Y. (1998). Topic detection and tracking pilot study final report.

[5] Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, *2*, 231-244.

[6] Kalloubi, F., & Nfaoui, E. H. (2016). Microblog semantic context retrieval system based on linked open data and graph-based theory. *Expert Systems with Applications*, *53*, 138-148.

[7] Shen, W., Wang, J., & Han, J. (2015). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, *27*(2), 443-460.

[8] Baldwin, T., Kim, Y. B., de Marneffe, M. C., Ritter, A., Han, B., & Xu, W. (2015). Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. *ACL-IJCNLP*, *126*, 2015.

[9] Weng, J., & Lee, B. S. (2011). Event detection in twitter. *ICWSM*, *11*, 401-408.

[10] Yang, X., Ghoting, A., Ruan, Y., & Parthasarathy, S. (2012, August). A framework for summarizing and analyzing twitter feeds. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 370-378). ACM.

[11] Kumaran, G., & Allan, J. (2004, July). Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 297-304). ACM.

[12] Hulpuş, I., Prangnawarat, N., & Hayes, C. (2015, October). Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In *International Semantic Web Conference* (pp. 442-457). Springer International Publishing.

[13] Vicient, C., & Moreno, A. (2015). Unsupervised topic discovery in micro-blogging networks. *Expert Systems with Applications*, *42*(17), 6472-6485.

[14] Decker, S., & Frank, M. (2004). The social semantic desktop. *Digital Enterprise Research Institute, DERI Technical Report May*, *2*, 7.

[15] Ha-Thuc, V., Mejova, Y., Harris, C., & Srinivasan, P. (2009). Event intensity tracking in weblog collections. *ICWSM, San Jose, USA*.

[16] Hepp, M. (2010). HyperTwitter: collaborative knowledge engineering via twitter messages. *Knowledge Engineering and Management by the Masses*, 451-461.

[17]. Gao, M., Liu, J., Zhong, N., Chen, F., & Liu, C. (2011). Semantic mapping from natural language questions to OWL queries. *Computational Intelligence*, *27*(2), 280-314.

[18] Aroyo, L., & Houben, G. J. (2010). User modeling and adaptive Semantic Web. *Semantic Web*, *1*(1, 2), 105-110.

[19] Gruhl, D., Nagarajan, M., Pieper, J., Robson, C., & Sheth, A. (2009). Context and domain knowledge enhanced entity spotting in informal text. *The Semantic Web-ISWC 2009*, 260-276.

[20] O'Shea, J., Bandar, Z., Crockett, K., & McLean, D. (2010). Benchmarking short text semantic similarity. *International Journal of Intelligent Information and Database Systems*, *4*(2), 103-120.

[21] Tsatsaronis, G., Varlamis, I., & Vazirgiannis, M. (2010). Text relatedness based on a word thesaurus. *Journal of Artificial Intelligence Research*, *37*, 1-39.

[22] Kumar, N. S., & Muruganantham, D. (2016). Disambiguating the Twitter Stream Entities and Enhancing the Search Operation Using DBpedia Ontology: Named Entity Disambiguation for Twitter Streams. *International Journal of Information Technology and Web Engineering (IJITWE)*, *11*(2), 51-62.

[23] Kalloubi, F., & Nfaoui, E. H. (2016). Microblog semantic context retrieval system based on linked open data and graph-based theory. *Expert Systems with Applications*, *53*, 138-148.

[24] Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., .& Bontcheva, K. (2015). Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, *51*(2), 32-49.

[25] El-Halees, A., & Al-Asmar, A. (2017). Ontology Based Arabic Opinion Mining. *Journal of Information & Knowledge Management*, *16*(03), 1750028.

[26] Lima, R., Espinasse, B., & Freitas, F. (2017). OntoILPER: an ontology-and inductive logic programming-based system to extract entities and relations from text. Knowledge and Information Systems, 1-33.

[27] Rehage, G., Joppen, R., & Gausemeier, J. (2016). "Perspective on the Design of a Knowledge-based System Embedding Linked Data for Process Planning". Procedia Technology, 26 ( 2016 ) 267 – 276.

[28] Otegi, A., Arregi, X., Ansa, O., & Agirre, E. (2015). Using knowledge-based relatedness for information retrieval. Knowledge and Information Systems, 44(3), 689-718.

[29] Madani, A., Boussaid, O., & Zegour, D. E. (2015). New Information in Trending Topics of Tweets by Labelled Clusters. *Journal of Information & Knowledge Management*, *14*(03), 1550019.

[30] Lima, R., Espinasse, B., & Freitas, F. (2017). OntoILPER: an ontology-and inductive logic programming-based system to extract entities and relations from text. Knowledge and Information Systems, 1-33.

[31] Izhar, T. A. T., Torabi, T., & Bhatti, M. I. (2017). Using Ontology to Incorporate Social Media Data and Organizational Data for Efficient Decision-Making. *International Journal of Computer Information Systems and Industrial Management Applications*, *9*(2017), 9-22.

[32] Kitajima, R., Kamimura, R., Uchida, O., & Toriumi, F. (2016). Potential information maximization: Potentiality-driven information maximization and its application to Tweets classification and interpretation. *International Journal of Computer Information Systems and Industrial Management Applications*, *8*, 42-51.

[33] Okoye, K., Tawil, A. R. H., Naeem, U., & Lamine, E. (2016). Discovery and enhancement of learning model analysis through semantic process mining. *International Journal of Computer Information Systems and Industrial Management Applications, 8*(2016), 93-114.

[34] Ali, F., & Shima, Y. (2016, November). On analysis and visualization of Twitter data. In *International Conference on Hybrid Intelligent Systems* (pp. 271-280). Springer, Cham.

## Author Biographies

Prof. SenthilKumar N received his Master Degree in M.Tech – IT from VIT University, Vellore and currently working as Assistant Professor in VIT University, Vellore, India. He has pocketed 10 years of teaching experience and his research areas includes Semantic Web, Information Retrieval and Web Services. He is currently holding a project on semantic understanding of named entities in the web and building a project on it.

Dr. Dinakaran M received his Doctorate in Computer Science from Anna University, Chennai and Master Degree in M.Tech IT from VIT University, Vellore. He is currently working as Associate Professor in VIT University, Vellore, India. He has good teaching experience of more than 8 years. His area of research includes Information Retrieval, Networking and Web Service Management..