Report On
Collecting social media data for planned events

submitted by
Akshay Avinash Jadhav(ES12B1004), Ratnesh Chandak(CS12B1030)

ABSTRACT

Twitter has become tremendously popular with hundreds of millions of tweets being posted every day on a variety of topics. Corporations are increasingly monitoring Twitter for gaining insights on entities of interest, such as events. Planned events are happenings or occurrences that happen at predefined places and predefined times, and are of interest to several people. Examples of such events are music concerts, sports matches, conferences etc. Often people discuss about these events in social media**.** The twitter search engine ranks the results to a user query in chronological order. With the ever increasing conversation on Twitter, it is difficult to get the relevant themes pertaining to an event. In this work, we have made a pipeline to extract the relevant themes for an event from Twitter using the given event metadata.

INTRODUCTION

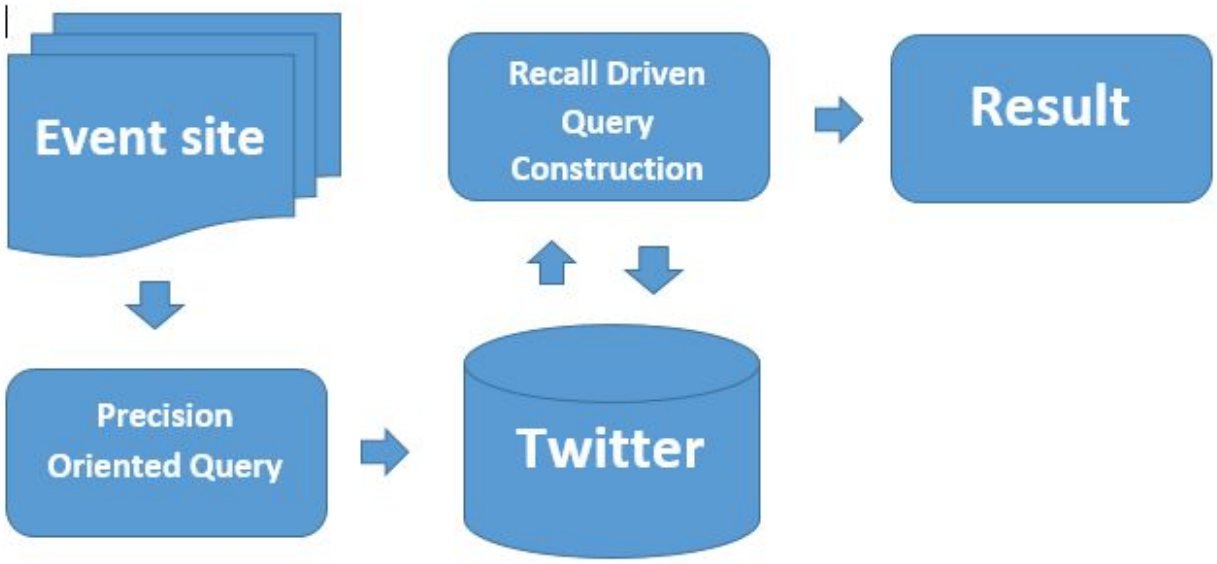
Every second, on average, around 6,000 tweets are tweeted on Twitter, which corresponds to over 350,000 tweets sent per minute, 500 million tweets per day and around 200 billion tweets per year. There are hundreds of tweets for any event, as a user one cannot scroll to all the tweets to get useful information of the event. There is information of very little things in the tweets but one has to define a system such that he can retrieve useful information from those tweets. The aim of our work is to get tweets which are relevant and which are information rich of the event. We leverage the explicit event details in the event metadata like the title of the event eg.:(“SunBurn Goa 2015”), city the event is taking place at eg.:(“Goa”), venue the event is happening at eg.:(“Vagator Beach”) ,date on which the event will take place eg.:(27th December, 2015) to automatically formulate precision queries. The results we get from the precision queries are of high precision and we use this high precision results to formulate recall oriented queries using text processing techniques like term extraction, frequency analysis and temporal analysis. Our contributions are as follows:

- We use twitter APIs to search tweets for an event using the precision oriented queries.
- We extract bigrams, trigrams, hashtags and URLs mentioned in the tweets.
- We perform frequency analysis to get the most frequent bigrams, trigrams, hashtags.
- We do a temporal analysis to get more relevant terms.

By getting the top bigrams, trigrams, hashtags and URLs, we greatly enhance the user's event based information seeking experience.

PROBLEM AND APPROACH

Problem: Given details of an event, goal of the project is to find out tweets about the event from the Twitter social media.
Approach: Our approach to the above problem is in two steps. First, we use precision oriented queries to get highly precision oriented results. Then too improve the recall of these results, we aim to formulate the recall oriented query. For this we form a set of candidate queries for our recall oriented queries which would retrieve additional tweets for the event.



Flow diagram showing our approach

PRECISION ORIENTED QUERIES

Our first step towards retrieving relevant tweets, is to get tweets that are high in precision. The tweets we get after searching with the precision oriented queries consists of the event related information. We use the precision oriented queries described by Hila Becker[1] as given in the below table.

Strategy	Example
[“title” + “city”]	“Sunburn” “Goa”
[title + “city”]	Sunburn “Goa”
[title - stopwords + “city”]	Sunburn “Goa”
[“title”+”venue”]	“Sunburn” “Vagator Beach”
[title + “venue”]	Sunburn “Vagator Beach”
[“title”]	“Sunburn”
[title]	Sunburn
[title - stopwords]	Sunburn

We get the title, city, venue from the event metadata we collected from various sites. The results we get by performing search using these precision oriented queries are the high precision results.

ANALYSING THE TWEETS

By analysing the tweets, we find the top bigrams, trigrams, hashtags and URLs. The bigrams, trigrams and hashtags will be our candidates for recall oriented query formulation.

Extracting the hyperlinks:

We formulated a regex for a hyperlink in a tweet. As a tweet is limited to 140 characters, people put a short link in the tweet. We expand these short links to get the original link the people wanted to mention in the tweet. We do a frequency analysis of the hyperlinks we get the top hyperlinks for each event.

Extracting the hashtags:

Most of the tweets we got consisted of hashtags. A hashtag is a word starting with "#", eg:#SunburnGoa. We formulated a regex for hashtags in a tweet. We find the matches for this reges in the tweets we have collected using the precision oriented queries. Then we do a frequency analysis of these hashtags to get the top hashtags for an event.

Extracting the bigrams and trigrams:

The first step in extracting the n-grams was to remove the stopwords from each tweet. We found that the unigrams won't be much informative as pairs of words which are bigrams will be more meaningful. In the tweets we collected, a tweet consisted 7 words on an average. Given below is the flow, how to extract bigrams from a tweet.

Step 1	Input tweet	'Bazaar' is the 'Official @SunburnFestival Goa 2015 Anthem'! Out on Dec 11 @SpinninRecords @KSHMRmusic pic.twitter.com/wYBt6jQpHq
Step 2	Remove URLs	'Bazaar' is the 'Official @SunburnFestival Goa 2015 Anthem'! Out on Dec 11 @SpinninRecords @KSHMRmusic
Step 3	Remove hashtags	'Bazaar' is the 'Official @SunburnFestival Goa 2015 Anthem'! Out on Dec 11 @SpinninRecords @KSHMRmusic
Step 4	Remove User mentions	'Bazaar' is the 'Official Goa 2015 Anthem'! Out on Dec 11
Step 4	Remove stopwords and special characters.	Bazaar Official Goa 2015 Anthem Out Dec 11
Step 5	Extract bigrams	Bazaar Official, Official Goa, Goa 2015, 2015 Anthem, Anthem Out, Out Dec, Dec 11

RECALL ORIENTED QUERIES

Till now, we have got the bigrams and trigrams. These are the candidate queries for our recall oriented queries. To select the most promising from this set of candidate queries, we perform frequency analysis and temporal profile analysis to rank our queries.

Frequency analysis:
Through this technique, we aim to extract the most frequently used terms by the people while tweeting for an event. To select the most frequent n-grams, we compute the frequencies for the bigrams and trigrams we have extracted. We reject the n-grams which have frequency less than 1% of the total tweets collected for that event.
These most frequent n-grams will still have noise. For eg.: ["concert tonite"] is a general bigram and not an event specific bigram. Issuing hundreds of queries for each event is not scalable and could potentially introduce substantial noise, so we further need to reduce our candidate queries. We perform two methods:

Specificity:
Specificity ensures that we rank along, detailed queries higher than broad, general ones. Since we have used conjunctive query semantics, longer queries consisting of multiple terms, are more restrictive than shorter queries consisting of few terms. In our case, we found that most of the trigrams are super sets of the bigrams. In such case, we favour the trigram.

Temporal Profile Analysis:
The temporal profile analysis pertains to ranking according to the historical profile of the n-gram. A local spike in the frequency of the n-gram around the time of the event might serve as an indication that the n-gram is infact associated with the event. We keep the record of the frequency of n-gram per day. If we get a spike in the frequency around 2 days before and after the event date, we rank the n-gram higher.

After this, we would have the candidate queries for formulating our recall oriented queries.

EXPERIMENT:

We used Twitter4J[2] library to query twitter for tweets. We designed a schema for storing tweets, users, places and events. All the code was written in Java.

Planned Event Dataset:
We assembled a dataset of 35 events from various event aggregation sites like meraevents, Eventbrite, Eventful, Bookmyshow and Facebook events. We formed the metadata for each events and stored them in our database.

Collection of tweets for each event:

We started the collecting tweets for events from November 5, 2015. For each event in our database, we collected tweets for a one month period ranging from two weeks before the event started to two weeks after the event was over.

We wrote a java code to query twitter using Twitter4J library. In one request it gives only 100 results. So we wrote a while loop which remembers the previous query results' last tweet id and uses it as the starting tweet id for the next iteration's query.

As twitter limits the search results to only a week prior. So, we collected tweets for the events in our database every weekend, so we have tweet of the whole week at once.

Extracting URLs:

As mentioned earlier, that a tweet generally contains short URLs. So to extract the original URL, we needed to expand the short URLs. We formulated our own regex to detect URLs. Then we tried expanding the short URLs by opening HTTP connections. It worked well for a URL which is shortened once. But for a URL shortened multiple times, multiple HTTP connections were to be opened which took a long time. So, we proceeded by using http://expandurl.appspot.com/ [3] which provides GET request API and gives the expanded URL of the shortened URL we give as the GET request query.

For each event, we stored the expanded URLs in our database.

Extracting Hashtags:

We formulated our own regex to detect hashtags: #\\S+. For each event, we stored the hashtags and their frequency in our database.

Extracting bigrams and trigrams:

Bigrams and trigrams were extracted as mentioned earlier in the flow to extract a bigram. These were stored in our database along with their frequency.

Temporal Analysis was done by making a database with daily frequencies of the n-grams. Then using an SQL query, we found those n-grams having a local spike.

In the end, we had the remaining bigrams and trigrams as the candidate queries for recall query generation.

LIMITATIONS:

The biggest limitation to our work was that twitter limits the search results to a week before. So we could not properly test our candidate queries we had formed for recall. Also, twitter API limits search queries upto 15000 requests, after which one has to wait 15 minutes before issuing one more search request.

CONCLUSION:

Given the planned event metadata, we have made a pipeline which will show the top bigrams, trigrams, URLs and hashtags for the event. Also, the top bigrams and trigrams form the candidate queries for the recall oriented queries.

References:

1) http://www.cs.columbia.edu/~gravano/Papers/2012/wsdm12.pdf
2) http://twitter4j.org/en/index.html
3) http://expandurl.appspot.com/