Online Sub-event detection on Twitter data stream

Myself: Anujraaj Goyal, IIIT Allahabad
Mentor: Koustav Rudra
Professor: Dr. Niloy Ganguly

the project

The corpus consists of tweets from a specific domain talking about a single major event.

The objective is to identify the sub-events within this event

Summarising the corpus by generating a short description about these subevents

Codebase:-
https://github.com/goel42/online-subevent-detection-twitter.git

approach

- Sub-event Detection : The first step is to identify if a sub-event has occurred and if it has, what tweets comprise the sub-event

- Tweet Selection : The second step is to choose representative tweets for each subevent

The aggregation of these two steps lend us a set of tweets as a summary of the event

References

- Sumblr: continuous summarization of evolving tweet streams
  Conference: SIGIR

- Sub-Event Detection During Natural Hazards Using Features of Social Media Data
  Conference: WWW

Input

Dataset containing tweets

For sub-event detection:-

- A count of subevents required
- A count of popular words per subevent

For summarisation:-

- Word limit for the summary

Output

For sub-event detection:-

- Display the top ranking clusters, each cluster identifying one subevent
- Popular words used to describe the corresponding subevents
- Brief description of the event

Detecting the sub-event

2 techniques used:-

- Assuming, the given dataset is static
- Assuming, the tweets come one by one and are iterated upon in an online fashion

Static Dataset

- We first generate a vocabulary of unique words in the corpus
- Using this, we represent each tweet as a vector of tf-idf values for the corresponding words in the a given tweet
- A cluster class to denote a particular sub-event
- Single Pass Incremental Clustering algorithm
- Clustering using only tweet text
- Clustering using tweet text and date-time of the tweet

Tf-idf vector calculation

- **TF: Term Frequency**, which measures how frequently a term occurs in a document
  - TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).
- **IDF: Inverse Document Frequency**, which measures how important a term is, on the corpus level
  - IDF(t) = log_e(Total number of documents / Number of documents with term t in it).
  - The idea is to dampen the score of very frequent terms, and boost the score of infrequent terms.

Cluster class

- A collection of tweets that identify the subevent
- Each tweet is represented as a vector of the tf-idf values of its constituent words
- For textual similarity, cosine similarity metric is used to measure the similarity between two tweets
- For date-time feature,
    - if difference in minutes < 720 (ie half day) , similarity is 1 - (minute_difference)/1440
    - If difference in minutes > 720 (ie half day), similarity is 0

Single-Pass Incremental Clustering Algorithm

Based on tweet text only

Each tweet is analysed and compared with the centroid of existing clusters, and the cluster with maximum similarity is noted

If the maximum similarity is greater than a preset threshold, then this tweet is assumed to belong to that cluster and is added to it, else a new cluster is generated

In the end, all the tweets are thus partitioned into clusters representing different subevents

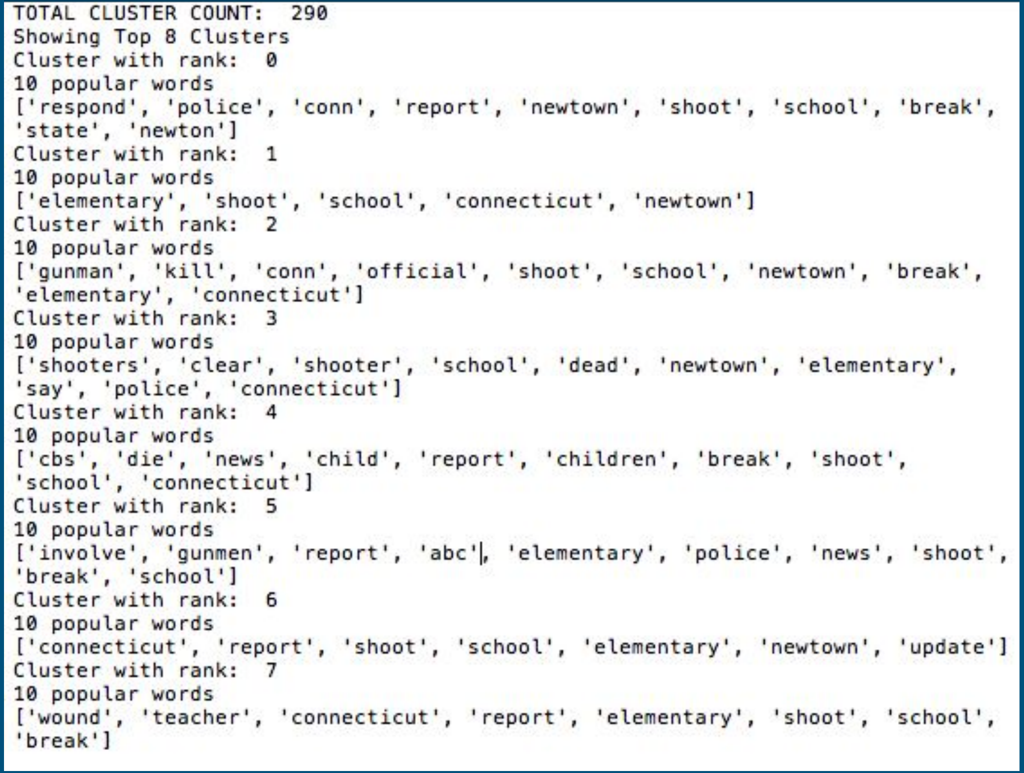
Single-Pass Incremental Clustering Algorithm

Based on text and date-time of tweet

A similar set of clusters based on date-time of the tweets, using the date-time similarity metric discussed previously,

The final set of clusters is generated using weighted binary voting of these two cluster lists

Results



```
TOTAL CLUSTER COUNT:  290
Showing Top 8 Clusters
Cluster with rank:  0
10 popular words
['respond', 'police', 'conn', 'report', 'newtown', 'shoot', 'school', 'break',
'state', 'newton']
Cluster with rank:  1
10 popular words
['elementary', 'shoot', 'school', 'connecticut', 'newtown']
Cluster with rank:  2
10 popular words
['gunman', 'kill', 'conn', 'official', 'shoot', 'school', 'newtown', 'break',
'elementary', 'connecticut']
Cluster with rank:  3
10 popular words
['shooters', 'clear', 'shooter', 'school', 'dead', 'newtown', 'elementary',
'say', 'police', 'connecticut']
Cluster with rank:  4
10 popular words
['cbs', 'die', 'news', 'child', 'report', 'children', 'break', 'shoot',
'school', 'connecticut']
Cluster with rank:  5
10 popular words
['involve', 'gunmen', 'report', 'abc', 'elementary', 'police', 'news', 'shoot',
'break', 'school']
Cluster with rank:  6
10 popular words
['connecticut', 'report', 'shoot', 'school', 'elementary', 'newtown', 'update']
Cluster with rank:  7
10 popular words
['wound', 'teacher', 'connecticut', 'report', 'elementary', 'shoot', 'school',
'break']
```

- Clusters are ranked based on the count of constituent tweets and displayed
- Popular words are ranked based on the tf-idf values of the cluster representative

Observations

- Threshold vs Count of subevents in a test dataset of ~1900 tweets:-

Threshold	Count of subevents
0.55	224
0.7	290
0.85	357

- In our dataset, the time interval between the tweets depicting different sub-events is very brief and non uniform, the quality of clusters formed isn't greatly affected by using the date-time feature.

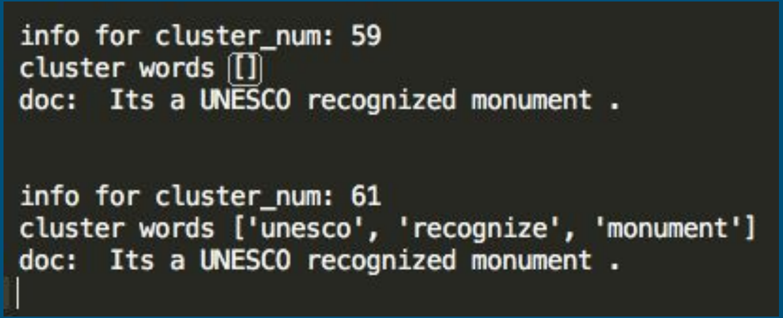
Online Method

In contrast to the static method, the tweets are not known beforehand. Consequently, the existing vocabulary (using which the tf-idf values are calculated) needs to be updated whenever the incoming tweet contains new words.

Each word in the vocabulary represents a dimension in the cluster representative vector, therefore after updating the vocabulary, the cluster representatives of all the previously generated clusters also need to be updated on the go.

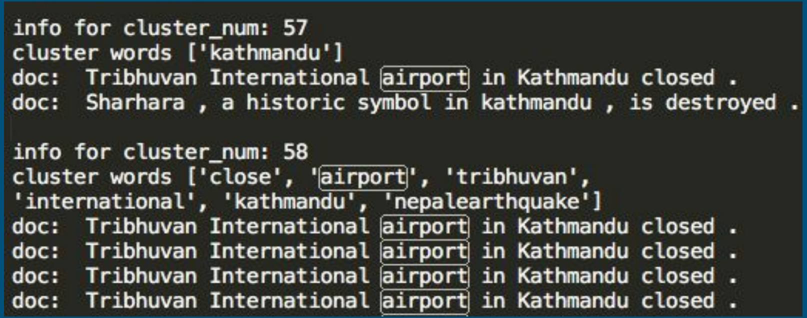
Observations

- If the words in an incoming tweet are new and not recognised in the vocabulary so far, identical tweets land in different clusters.
- If the words in the incoming tweets are entirely new, clusters with empty representatives are formed.



```
info for cluster_num: 59
cluster words []
doc:  Its a UNESCO recognized monument .


info for cluster_num: 61
cluster words ['unesco', 'recognize', 'monument']
doc:  Its a UNESCO recognized monument .
```



```
info for cluster_num: 57
cluster words ['kathmandu']
doc:  Tribhuvan International airport in Kathmandu closed .
doc:  Sharhara , a historic symbol in kathmandu , is destroyed .

info for cluster_num: 58
cluster words ['close', 'airport', 'tribhuvan',
'international', 'kathmandu', 'nepalearthquake']
doc:  Tribhuvan International airport in Kathmandu closed .
doc:  Tribhuvan International airport in Kathmandu closed .
doc:  Tribhuvan International airport in Kathmandu closed .
doc:  Tribhuvan International airport in Kathmandu closed .
```

We handle this using a merge step discussed in slides ahead

Merge step

The cluster set obtained so far are is considerably large.
Ex: ~1200 clusters for a corpus of ~10k tweets

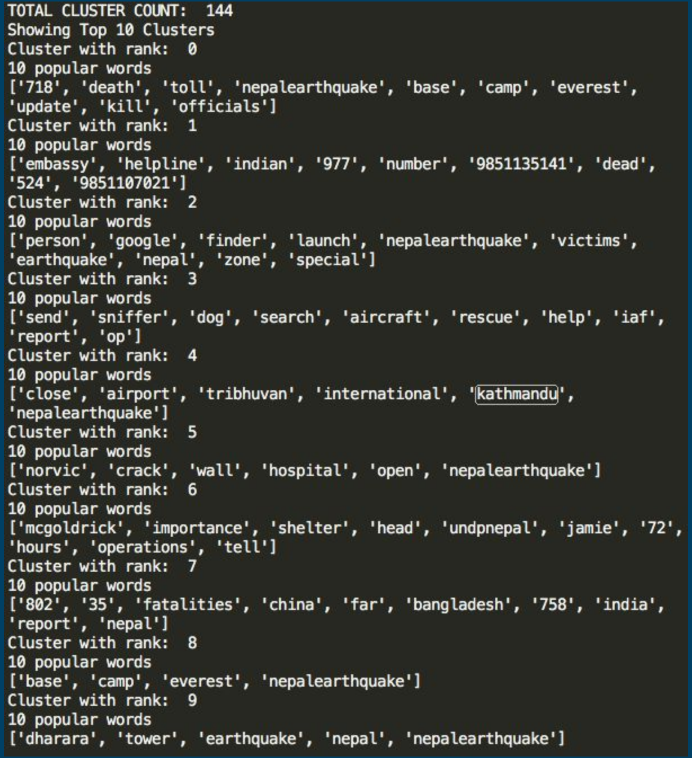
We compress the cluster set using a merging operation whenever the cluster count becomes greater than an upper limit N_{max}. This step lends us clusters that more cohesive and maintains an upper limit on the total number of clusters formed

Cluster merging Algorithm

- First, we sort all cluster pairs by their centroid similarities in a descending order.
- Beginning with the most similar pair, we try to merge two clusters in the pair.
- When both clusters are single clusters which have not been merged with other clusters, they are merged into a new composite cluster.
- When one of them belongs to a composite cluster (it has been merged with others before), then the other is also merged into that composite cluster.
- When both of them have been merged, if they belong to the same composite cluster, this pair is skipped
- otherwise, the two composite clusters are merged together.

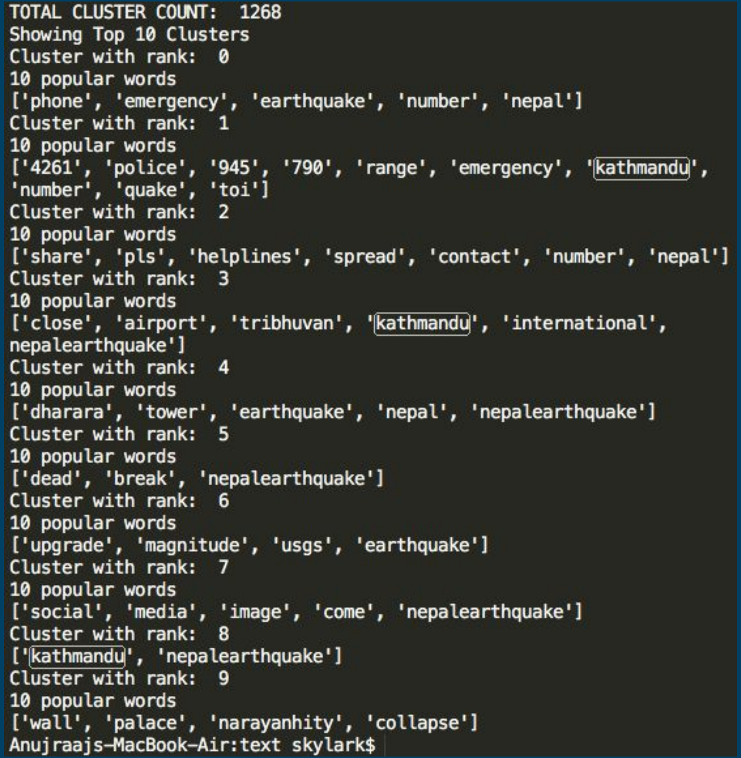
This process continues until there are only mc percentage of the original clusters left.

Results



```
TOTAL CLUSTER COUNT:  144
Showing Top 10 Clusters
Cluster with rank:  0
10 popular words
['718', 'death', 'toll', 'nepalearthquake', 'base', 'camp', 'everest',
'update', 'kill', 'officials']
Cluster with rank:  1
10 popular words
['embassy', 'helpline', 'indian', '977', 'number', '9851135141', 'dead',
'524', '9851107021']
Cluster with rank:  2
10 popular words
['person', 'google', 'finder', 'launch', 'nepalearthquake', 'victims',
'earthquake', 'nepal', 'zone', 'special']
Cluster with rank:  3
10 popular words
['send', 'sniffer', 'dog', 'search', 'aircraft', 'rescue', 'help', 'iaf',
'report', 'op']
Cluster with rank:  4
10 popular words
['close', 'airport', 'tribhuvan', 'international', 'kathmandu',
'nepalearthquake']
Cluster with rank:  5
10 popular words
['norvic', 'crack', 'wall', 'hospital', 'open', 'nepalearthquake']
Cluster with rank:  6
10 popular words
['mcgoldrick', 'importance', 'shelter', 'head', 'undpnepal', 'jamie', '72',
'hours', 'operations', 'tell']
Cluster with rank:  7
10 popular words
['802', '35', 'fatalities', 'china', 'far', 'bangladesh', '758', 'india',
'report', 'nepal']
Cluster with rank:  8
10 popular words
['base', 'camp', 'everest', 'nepalearthquake']
Cluster with rank:  9
10 popular words
['dharara', 'tower', 'earthquake', 'nepal', 'nepalearthquake']
```

After merge operation



```
TOTAL CLUSTER COUNT:  1268
Showing Top 10 Clusters
Cluster with rank:  0
10 popular words
['phone', 'emergency', 'earthquake', 'number', 'nepal']
Cluster with rank:  1
10 popular words
['4261', 'police', '945', '790', 'range', 'emergency', 'kathmandu',
'number', 'quake', 'toi']
Cluster with rank:  2
10 popular words
['share', 'pls', 'helplines', 'spread', 'contact', 'number', 'nepal']
Cluster with rank:  3
10 popular words
['close', 'airport', 'tribhuvan', 'kathmandu', 'international',
nepalearthquake']
Cluster with rank:  4
10 popular words
['dharara', 'tower', 'earthquake', 'nepal', 'nepalearthquake']
Cluster with rank:  5
10 popular words
['dead', 'break', 'nepalearthquake']
Cluster with rank:  6
10 popular words
['upgrade', 'magnitude', 'usgs', 'earthquake']
Cluster with rank:  7
10 popular words
['social', 'media', 'image', 'come', 'nepalearthquake']
Cluster with rank:  8
['kathmandu', 'nepalearthquake']
Cluster with rank:  9
10 popular words
['wall', 'palace', 'narayanhity', 'collapse']
Anujraajs-MacBook-Air:text skylark$
```

Without merge operation

Part 2: Summarisation

We use a greedy algorithm to select representative tweets (focus set) from each cluster to form summaries.This focus set contains ‘k’ tweets in the cluster that are nearest to the corresponding cluster representative.

Our summarisation tends to select tweets such that it can cover most information of the whole tweet set. This set of tweets provides us a brief description of main event in the corpus.

Summarisation Algorithm

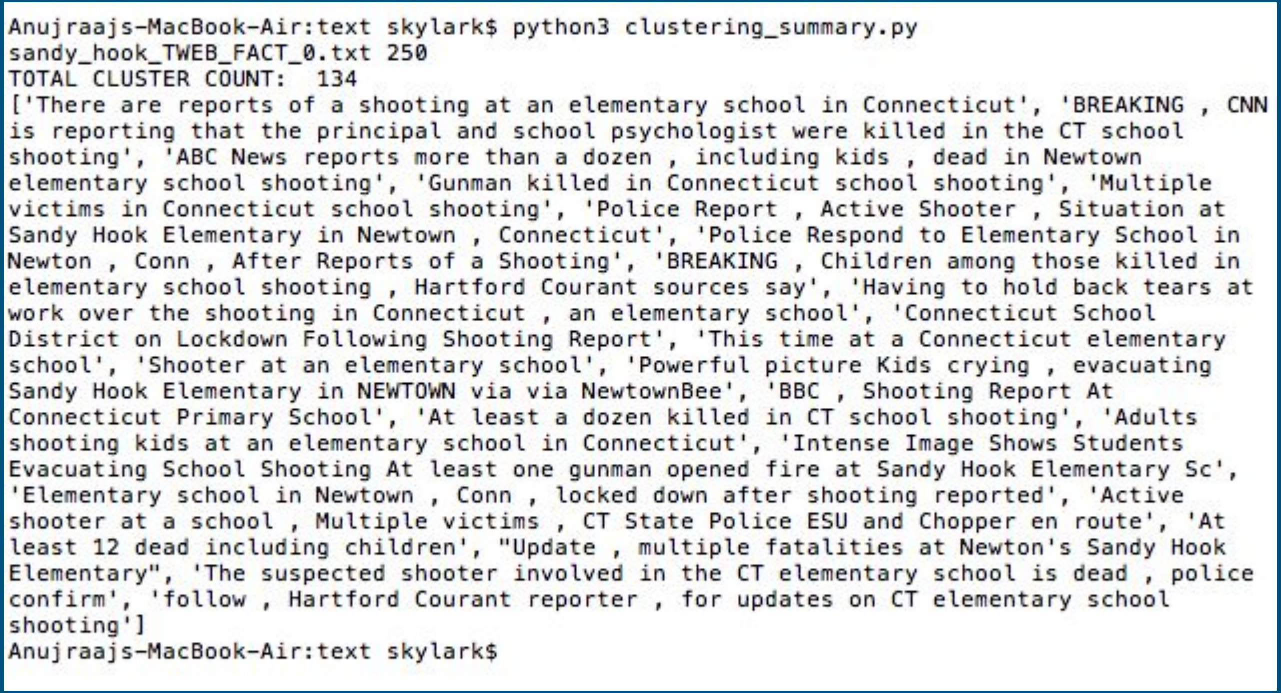
From the focus set of tweets, we generate a cosine-similarity graph. In this graph, a tweet from the focus set represents a node and tweets this similarity greater than a minimum threshold are connected by an edge

We obtain a set of tweet centrality scores using the PageRank algorithm, based on this graph that it takes as an input.

However, a potential problem of LexRank is that some top-ranked tweets may have similar contents. So, we choose one tweet with the highest LexRank score from each cluster which is sufficiently unique(based on a threshold value) in the summary generated so far.

The iterations proceed until the word limit is reached.

Results



```
Anujraajs-MacBook-Air:text skylark$ python3 clustering_summary.py
sandy_hook_TWEB_FACT_0.txt 250
TOTAL CLUSTER COUNT:  134
['There are reports of a shooting at an elementary school in Connecticut', 'BREAKING , CNN
is reporting that the principal and school psychologist were killed in the CT school
shooting', 'ABC News reports more than a dozen , including kids , dead in Newtown
elementary school shooting', 'Gunman killed in Connecticut school shooting', 'Multiple
victims in Connecticut school shooting', 'Police Report , Active Shooter , Situation at
Sandy Hook Elementary in Newtown , Connecticut', 'Police Respond to Elementary School in
Newton , Conn , After Reports of a Shooting', 'BREAKING , Children among those killed in
elementary school shooting , Hartford Courant sources say', 'Having to hold back tears at
work over the shooting in Connecticut , an elementary school', 'Connecticut School
District on Lockdown Following Shooting Report', 'This time at a Connecticut elementary
school', 'Shooter at an elementary school', 'Powerful picture Kids crying , evacuating
Sandy Hook Elementary in NEWTOWN via via NewtownBee', 'BBC , Shooting Report At
Connecticut Primary School', 'At least a dozen killed in CT school shooting', 'Adults
shooting kids at an elementary school in Connecticut', 'Intense Image Shows Students
Evacuating School Shooting At least one gunman opened fire at Sandy Hook Elementary Sc',
'Elementary school in Newtown , Conn , locked down after shooting reported', 'Active
shooter at a school , Multiple victims , CT State Police ESU and Chopper en route', 'At
least 12 dead including children', "Update , multiple fatalities at Newton's Sandy Hook
Elementary", 'The suspected shooter involved in the CT elementary school is dead , police
confirm', 'follow , Hartford Courant reporter , for updates on CT elementary school
shooting']
Anujraajs-MacBook-Air:text skylark$
```

Parameter Values

- Cosine Similarity threshold = 0.7
- Binary weights = [0.70, 0.30]
- Final Binary voting threshold = 0.5
- Count of Tweets in focus set of each cluster = 40
- Merge Coefficient = 0.7
- Cosine similarity graph threshold = 0.2
- PageRank Convergence Threshold = 0.0001
- Summarisation Repetition Threshold = 0.4

Further Work

Preprocessing the corpus so as to replace synonymous terms with a common value.

Focus set contains multiple identical tweets, which prevents fresh incoming tweets from coming in

Make a summary in a chronological order/ timeline fashion.

TASK 2: Determining optimum number of topics for Latent Dirichlet Allocation (LDA)

- LDA is one of methods to discover hidden topics in a text collection . The hidden topics are represented by word and document distribution over topics.

- The hidden topics are represented by word and document distribution over topics. This document distribution is used for assigning documents into clusters

- We use a cluster validation approach to identify the most suitable value of k for topical clusters
- Given as input is a range of values in which ‘k’ is expected to lie. This cluster validation finds a suitable value of ‘k’ such that the clusters are stable.

- Suppose that T is a set of tweets, and T‘ is a subset of T. Given a pair of tweets belonging to both T and T’, the clusters are stable if t1 and t2 are assigned into the same cluster when running LDA on T and T’
- Proportion of stability measures the stability of k clusters:

$$F_k(C', C) = \frac{\sum_{ij} 1(C'_{ij} = C_{ij} = 1, t_i, t_j \in T')}{\sum_{ij} 1(C_{ij} = 1, t_i, t_j \in T')}$$

Code:-

https://github.com/goel42/lda-optimum-topics.git

Thank you!

Suggestions? Remarks? Questions? Feedback?