Summarizing Microblogs During Emergency Events: A Comparison of Extractive Summarization Algorithms

Soumi Dutta, Vibhash Chandra, Kanav Mehra, Sujata Ghatak, Asit Kumar Das and Saptarshi Ghosh

Abstract Microblogging sites, notably Twitter, have become important sources of real-time situational information during emergency events. Since hundreds to thousands of microblogs (tweets) are generally posted on Twitter during an emergency event, manually going through every tweet is not feasible. Hence, summarization of microblogs posted during emergency events has become an important problem in recent years. Several summarization algorithms have been proposed in the literature, both for general document summarization, as well as specifically for summarization of microblogs. However, to our knowledge, there has not been any systematic analysis on which algorithms are more suitable for summarization of microblogs posted during disasters. In this work, we evaluate and compare the performance of 8 extractive summarization algorithms in the application of summarizing microblogs posted during emergency events. Apart from comparing the performances of the algorithms, we also find significant differences among the summaries produced by different algorithms over the same input data.

Keywords Summarization · Twitter · Microblogs · Extractive · Comparison Evaluation · Rouge

S. Dutta (✉) · V. Chandra · K. Mehra · A. K. Das · S. Ghosh
Indian Institute of Engineering Science and Technology Shibpur, Shibpur, India
e-mail: soumi.it@gmail.com

V. Chandra
e-mail: vibhashchandra2010@gmail.com

K. Mehra
e-mail: kanav.mehra6@gmail.com

A. K. Das
e-mail: akdas@cs.becs.ac.in

S. Dutta · S. Ghatak
Institute of Engineering and Management, Kolkata 700091, India
e-mail: ghatak.sujata07@gmail.com

S. Ghosh
Indian Institute of Technology Kharagpur, Kharagpur 721302, India
e-mail: saptarshi.ghosh@gmail.com

A. Abraham et al. (eds.), *Emerging Technologies in Data Mining and Information Security*, Advances in Intelligent Systems and Computing 813,
https://doi.org/10.1007/978-981-13-1498-8_76

1 Introduction

Microblogging sites like Twitter have become extremely important sources of real-time information on ongoing events, including socio-political events, natural and man-made emergencies, and so on. Especially during emergency events, such as disasters, microblogging sites are very important sources of situational information [1]. During such events, microblogs are usually posted so rapidly, and in such large volumes, that it is not feasible for human users to go through all the posts. In such scenario, it is critical to summarize the microblogs (tweets) and present informative summaries to the people who are attempting to respond to the disaster.

Automatic document summarization is a well-established problem in Information Retrieval, and many algorithms have been proposed for the problem. The reader is referred to [2, 3] for surveys on summarization algorithms. Summarization methods are broadly of two types—abstractive and extractive. While extractive algorithms generate summaries by extracting certain portions of the input data (e.g. certain sentences that are deemed important), abstractive algorithms attempt to generate summaries by paraphrasing parts of the input data. Out of these, a majority of the algorithms proposed in literature are extractive in nature [3].

With the recent popularity of microblogs as a source of information, a number of summarization algorithms have been recently proposed specifically for microblogs as well (see Sect. 2 for details). The problem of microblog summarization is inherently a multi-document summarization problem. However, algorithms for single- document summarization are also applicable, by considering the input set of microblogs to make up a single document. Microblog summarization has some distinct challenges, primarily due to the small size of individual microblogs, and the noisy, informal nature of microblogs, which make it difficult to interpret the semantic similarity of microblogs.

Several summarization algorithms exist in the literature, both for general documents, as well as specifically for microblogs. However, to our knowledge, there has not been any systematic analysis on how effective these algorithms are in the application of summarizing microblogs posted during disaster events. In this work, we evaluate and compare *eight* off-the-shelf extractive summarization algorithms for the said application. We perform experiments over microblogs related to five recent disaster events. We observe that different off-the-shelf algorithms generate vastly different summaries from the same input set of microblogs, with very few tweets in common between the summaries generated by different algorithms. Additionally, we evaluate the performance of the different algorithms using the standard ROUGE measure. We observe that the LUHN [4] and COWTS [5] algorithms achieve relatively high ROUGE scores, as compared to the other algorithms considered here.

The rest of the paper is organized as follows. A short literature survey on summarization of microblogs is presented in Sect. 2. Section 3 describes the summarization algorithms chosen for the comparative analysis. The microblog datasets used for the comparison are described in Sect. 4, while Sect. 5 discusses the results of the comparison among the various algorithms. The paper is concluded in Sect. 6, which also states some potential future research directions.

2 Related Work

A large number of document summarization algorithms have been proposed in the literature. The reader can refer to [2, 3] for surveys on summarization algorithms. Since the present work is specifically on summarization of microblogs/tweets, we focus on summarization algorithms for microblogs in this section.

2.1 Summarization of Microblogs

Several algorithms for microblog summarization have been proposed in recent years [6–9]. For instance, Shou et al. [7] propose a system based on initially clustering similar tweets and then selecting few representative tweets from each cluster, finally ranking these according to importance via a graph-based approach (LexRank) [10]. Extracting bigrams from the tweets are considered as the graph-nodes, Olariu [6] proposed a graph-based abstractive summarization System. Some other authors have also proposed graph-based methods for summarization of tweets [11, 12].

Some other works have proposed methodologies to summarize microblogs posted during specific events, such as sports events [13–17]. Considering greedy summarization Osborne et al. [18] proposed a real event tracking system.

Along with general microblog summarization approaches, a few recent studies have also focused specifically on summarization of news articles and tweets posted during emergency events [5, 19, 20]. In particular, our prior work [5] proposed a classification–summarization technique to extract and summarize situational information from tweet streams. An approach based on Integer Linear Programming was applied to optimize for the presence of certain important terms (called “content words”) in the summary.

2.2 Prior Works on Comparing Algorithms for Microblog Summarization

There have been two notable works on comparing different algorithms for microblog summarization [21, 22].

In a study, Mackie et al. [21] evaluated the performance comparison of 11 summarization approaches for 4 microblog datasets. The 11 summarization methods include a random baseline, temporal approaches (e.g. those that rank tweets by time), approaches based on term statistics such as *tf-idf*, and approaches based on term statistics, novelty, and cohesiveness [23].

Inouye et al. [22] compared algorithms for extractive summarization of Twitter posts. Two types of algorithms were considered which select tweets to produce summaries from a given set—(i) hybrid *tf-idf*-based algorithm, and (ii) a clustering-based algorithm. The performances of these algorithms are compared with manually produced summaries (gold standard) and summaries produced by eight different summarizers such as random, most recent, MEAD, TextRank, LexRank, cluster, Hybrid TF-IDF and SumBasic. The comparison showed that frequency- based summarizers (hybrid TF-IDF and SumBasic) achieve the best results in terms of ROUGE F-score.

The present work is different from these prior works as follows. First, unlike the other works, we focus on summarization of microblogs posted during disasters, which is a problem of practical interest. Second, neither of the prior works compare as many off-the-shelf summarization algorithms (eight) as we do in this work. Additionally, we also check the overlap between the summaries generated by various algorithms—which none of the prior works have done.

3 Summarization Algorithms

In this section, we outline the extractive summarization algorithms that we considered for comparison in the present work. Note that some of these algorithms were originally proposed for summarization of a single document, where the sentences of the given document are ranked according to some importance measure, and then few important sentences are selected for the summary. These algorithms can be easily applied to summarization of a set of microblogs, where each microblog is analogous to a sentence.

(1) ClusterRank: ClusterRank [24] is an unsupervised, graph-based approach which was originally designed for extractive summarization of meeting transcripts. ClusterRank algorithm is the extension of another algorithm named TextRank, which is also a graph-based method for extracting sentences from news articles. ClusterRank first segments the transcript into clusters which are represented as nodes in a graph. The similarity between all pairs of adjacent clusters is then measured, and the pair with the highest similarity is merged into a single cluster. Following this, a centroid-based approach is used to score each sentence within an important cluster. Relevancy of the sentences is also measured in addition to handling ill-formed sentences with high redundancy. Finally, the algorithm selects the highest scoring sentence and includes sentences in the summary until the length constraint is satisfied.

(2) COWTS: COWTS [5] is specifically designed for summarizing microblogs that are posted during disaster situations. The proposed approach is a classification–summarization framework for extracting meaningful situational information from microblog streams posted during disaster scenarios. Using low-level lexical and syntactic features, the classifier first distinguishes between situational and

non-situational information. Due to use of vocabulary-independent features, the classifier functions accurately in cross-domain scenarios. In COWTS, initially the classifier is trained over tweets posted during earlier disaster events and then deployed on tweets posted during a later disaster event. Then, the situational tweet stream is summarized by optimizing for the coverage of important *content words* (nouns, verbs, numerals) in the summary, using an Integer Linear Programming (ILP) framework.
(3) Frequency Summarizer: This is a simple summarization algorithm, which attempts to extract a subset of sentences which cover the main topics of a given document. The algorithm works on the simple idea that sentences which contain the most recurrent words in the text, are likely to cover most of the topics of the text.
(4) LexRank: LexRank [25] is stochastic graph-based method for computing relative importance of textual units in a document. In this method, a graph is generated which is composed of all sentences in the input corpus. Each sentence is represented as a node, and the edges denote similarity relationships between sentences in the corpus. An intra-sentence cosine similarity measure is used as edge weight in the graph representation of sentences by considering every sentence as bag-of-words model. A connectivity matrix or similarity matrix is generated using the similarity measure, which can be used as a similarity graph between sentences. A thresholding mechanism is applied (i.e. edges having weights below the threshold are removed) to extract the most important sentences from the resulting similarity matrix. A scheme based on Eigenvector centrality is also used to rank the sentences (nodes). The sentences are then included in the summary based on their importance values.
(5) LSA: LSA [26] is a generic extractive text summarization method to identify semantically important sentences for generating the summary. It is an unsupervised method of deriving vector space semantic representations from large documents, and does not need any training or external knowledge. Considering context of the input document, LSA extracts information such as which words are used together and which common words are seen in different sentences. High number of common words among sentences means that the sentences are more semantically related. LSA is based on mathematical technique which is named Singular Value Decomposition (SVD) [27] that is used to find out the interrelations between sentences and words. The input text document is first converted into a matrix, where each row represents a word and each column represents a sentence. Each cell value represents the importance of the word. SVD is then applied on this matrix to select the sentences to generate the summary.
(6) LUHN: Luhn’s algorithm [4] works on the perception that some words in a document are descriptive of its content, and the sentences that express the most significant information in the document are the ones that contain many such descriptive words close to each other. The words that occur often in a document are likely to be associated with the main topic of the document. However, an exception to this observation is stopwords. Hence, Luhn proposed the idea of stopwords such as determiners, prepositions and pronouns, which do not have much value in informing about the topic of the document. So he suggested removing these words from consideration. Luhn identified descriptive words using empirically determined high- and low-frequency thresholds. The high-frequency thresholds filter out the words that occur very

frequently throughout the article. Similarly, the low-frequency thresholds filter out the words that occur too infrequently. The remaining words in the document are the descriptive words, which indicate that content which is important.

On a sentence level, a ‘significance factor’ is computed for each sentence, which could be calculated for a sentence by bracketing the significant words in the sentence, squaring the number of significant words and then dividing by the total number of words. Sentences are identified as important and included in the summary based on the significance factor values.

(7) Mead: Mead [28, 29] is a centroid-based multi-document summarizer. First, topics are detected by agglomerative clustering that operates over the tf-idf vector representations of the documents. Second, a centroid-based methodology is used to identify sentences in each cluster that are central to the topic of the entire cluster. For each sentence, three different features are computed, which are its centroid value, positional value and first-sentence overlap. A composite score of each sentence is generated as a combination of the three scores. The score is further refined after considering possible cross-sentence dependencies, e.g. repeated sentences, chronological ordering, source preferences.) Sentences are finally selected based on this score.

(8) SumBasic: SumBasic [30] is a frequency-based multi-document summarizer. SumBasic uses a multinomial distribution function to compute the probability distribution over the words in a sentence. Based on average probability of occurrence of the words in the sentence, scores are assigned to each sentence. Then the sentences with the best scores are selected. Successively, the word probabilities and sentence scores are updated until the desired summary length is reached. The updation of word probabilities gives a natural way to deal with the redundancy in the multi-document input.

It can be noted that we selected the algorithms described above because either their implementations are readily available off the shelf, or they are easy to implement.[1]

4 Dataset for Comparison of Summarization Algorithms

This section describes the dataset we use to compare the various summarization algorithms.

[1] Availability of implementations: Frequency Summarizer (http://glowingpython.blogspot.in/2014/09/text-summarization-with-nltk.html), Mead (http://www.summarization.com/mead/), SumBasic (https://github.com/EthanMacdonald/SumBasic), LexRank, LSA and LUHN are available as part of the Python Sumy package (https://pypi.python.org/pypi/sumy). COWTS (proposed in our prior work [5]) and ClusterRank were implemented by us.

4.1 Emergency Events Considered

We considered tweets posted during the following emergency events.

1. **HDBlast—**two bomb blasts in the city of Hyderabad, India [31],
2. **SHShoot—**an assailant killed 20 children and 6 adults at the Sandy Hook elementary school in Connecticut, USA [32],
3. **UFlood—**devastating floods and landslides in the Uttaranchal state of India [33],
4. **THagupit—**a strong cyclone code named Typhoon Hagupit hit Philippines [34],
5. **NEquake—**a devastating earthquake in Nepal [35].

The dataset used for experimental purpose, are the selected events occurred during natural and man-made disasters in various regions of the world. Hence, the vocabulary/linguistic style in the tweets can be predictable to be dissimilar as well.

4.2 Developing the Dataset

For experiment, we have collected relevant tweets posted during each event through the Twitter API [36] using keyword-based matching. For instance, to identify tweets related to the HDBlast event the keywords such as 'Hyderabad', 'bomb' and 'blast' are used and to collect tweets related to the SHShoot event the keywords 'Sandyhook' and 'shooting' are considered.

We initially considered the chronologically earliest 1,000 tweets for each event. Due to frequent retweeted/re-posted by multiple users [37] Twitter often contains duplicates and near-duplicates tweets. Since such near-duplicates are not useful for the purpose of summarization, we removed them using a simplified version of the techniques suggested in [37], as follows.

Each tweet was considered as a bag of words (excluding standard English stopwords and URLs), and the similarity between two tweets was measured as the Jaccard similarity between the two corresponding bags (sets) of words. The two tweets were considered near-duplicates if the Jaccard similarity between two tweets was found

Table 1 Datasets used for the experiments. 1,000 chronologically earliest tweets were initially considered for each event, and near-duplicates were removed using methods in [37]. The last column shows the number of distinct tweets, after removing near-duplicates

Bomb blasts in the city of Hyderabad, India	95
Earthquake in Nepal in April 2015	146
Floods and landslides in the Uttaranchal state of India	173
Sandy Hook elementary school shooting in USA	252
Typhoon Hagupit in Philippines	484

to be higher than a threshold value (0.7) and only the longer tweet (potentially more informative) was retained. Table 1 shows the number of distinct tweets in each dataset after removal of duplicates and near-duplicates.

5 Experimental Results

We describe our experimental results in this section. Apart from comparing the performances of different algorithms, we also check whether different summarization algorithms produce very different summaries from the same input data.

5.1 *Do Different Algorithms Produce Very Different Summaries?*

Extractive summarisation algorithms for microblogs will select a subset of the tweets for inclusion in the summary. We first investigate whether different algorithms select a common set of tweets for the summaries, or whether the sets of tweets selected by different algorithms (for inclusion in summary) vary significantly.

Interestingly, we observed that *different summarization algorithms usually select very different sets of tweets in the summaries*. To demonstrate this observation, Table 2 shows the overlap between the summaries generated by the different algorithms, for the Nepal earthquake dataset. The entry (i, j), $1 \leq i, j \leq 8$ in Table 2 shows the number of common tweets included in the summaries generated by the two algorithms A_i and A_j. Similarly, Table 3 shows the overlaps for the Sandy Hook dataset.

Table 2 Overlap of tweets in the summaries (of length 25 tweets each) generated by different base summarization algorithms, for the Nepal earthquake dataset. Other datasets also show very low overlap

Algorithm	CR	CW	FS	LR	LS	LH	MD	SB
ClusterRank (CR)	–	7	4	0	3	2	4	4
COWTS (CW)	7	–	4	0	2	2	1	5
FreqSum (FS)	4	4	–	3	2	4	2	5
LexRank (LR)	0	0	3	–	1	2	0	2
LSA (LS)	3	2	2	1	–	7	3	4
LUHN (LH)	2	2	4	2	7	–	1	1
MEAD (MD)	4	1	2	0	3	1	–	0
Sumbasic (SB)	4	5	5	2	4	1	0	–

Table 3 Overlap of tweets in the summary (of length 25 tweets) generated by diffrent base summarization algorithms, for the Sandy Hook school shooting dataset

Algorithm	CR	CW	FS	LR	LS	LH	MD	SB
ClusterRank (CR)	–	1	0	0	2	1	2	4
COWTS (CW)	1	–	2	0	3	0	1	7
FreqSum (FS)	0	2	–	1	5	7	0	2
LexRank (LR)	0	0	1	–	1	2	0	0
LSA (LS)	2	3	5	1	–	13	3	3
LUHN (LH)	1	0	7	2	13	–	4	1
MEAD (MD)	2	1	0	0	3	4	–	1
Sumbasic (SB)	4	7	2	0	3	1	1	–

Table 4 Examples of tweets that were selected by at least four algorithms (out of eight) for inclusion in the summaries

Event	Tweet text
Hagupit	@EarthUncutTV: Latest 06z/2pm Philippines time JMA forecast track for #typhoon #Hagupit #RubyPH [url]
Hdblast	FLASH: 9 killed, 32 injured in serial powerful #blast in Dilshuknagar area in #Hyderabad: Police
SHshoot	Powerful picture RT @HeidiVoight Kids crying, evacuating Sandy Hook Elementary in NEWTOWN [url] via @BKnox88 via NewtownBee
UFlood	Really sad to hear news Uttarakhand floods my prayers with you Lord Shiva plz help them & plz take care n come back home Mumbai
UFlood	INDIAN ARMY IN FLOOD RELIEF OPERATIONS Uttarakhand Flood Helpline numbers 0135-2710334, 0135-2710335. [url]

It is evident from these tables that there is very low overlap between summaries generated by various base algorithms (similar trends are observed for all the datasets). The overlap is slightly higher in few specific cases, e.g. for the LUHN and LSA algorithms, possibly because these algorithms work on similar principles. However, most algorithms produce summaries that have very few tweets in common with summaries produced by other algorithms.

In spite of the low overlap between summaries produced by different algorithms, we find a few specific tweets which are selected by multiple algorithms for inclusion in the summaries. Table 4 shows examples of tweets that were selected at least four algorithms (out of eight) for inclusion in the summary. We observed that these tweets contain several terms which have high document frequency, i.e. terms which occur in many of the tweets in the particular dataset. As a result, these tweets are deemed (by multiple algorithms) to be good representatives of the whole dataset.

5.2 Evaluation of Summarization Algorithms

Next we focus on the evaluation of the performance of the different algorithms. To evaluate the quality of a summary (produced by an algorithm), we follow the standard procedure of generating gold standard summaries by human annotators, and then comparing the algorithm-generated summary with the gold standard ones. We employed three human annotators, each of whom is proficient in English and is a habitual user of Twitter, and has prior knowledge of working with social media content posted during disasters. Each annotator was asked to independently summarize each of the five datasets, and prepare extractive summaries of length 25 tweets each.

We executed all the summarization algorithms (described in Sect. 3) on each dataset, and obtain summaries of length 25 tweets each. We used the standard ROUGE measure [38] for evaluating the quality of the summaries generated by different algorithms, based upon their match with the gold standard summaries. Due to the informal nature of tweets, we considered the Recall and F-score of the ROUGE-1, ROUGE-2, and ROUGE-L variants.

Table 5 reports the performance of the different summarization algorithms, averaged over all the five datasets. We find that the LUHN algorithm performs the best for all the measures, followed closely by the COWTS algorithm.

To qualitatively demonstrate the differences between the summaries which obtain high ROUGE scores and those that obtain low ROUGE scores, Table 6 and Table 7, respectively, show the summaries generated by the LUHN algorithm (which obtained highest ROUGE score) and the LexRank algorithm (which obtained the lowest ROUGE score) for the same dataset—the Hyderabad blast dataset. It is evident that

Table 5 Performance of the summarization algorithms, averaged over all five datasets. The best performance is by the LUHN algorithm (highlighted in boldface) followed by the COWTS algorithm

Algorithm	Rouge-1		Rouge-2		Rouge-L	
	Recall	F-score	Recall	F-score	Recall	F-score
ClusterRank	0.459	0.467	0.230	0.233	0.448	0.456
COWTS	0.546	0.518	0.326	0.308	0.533	0.506
FreqSum	0.405	0.411	0.191	0.192	0.393	0.398
LexRank	0.278	0.371	0.124	0.164	0.273	0.365
LSA	0.515	0.486	0.284	0.267	0.503	0.475
LUHN	**0.563**	**0.531**	**0.331**	**0.313**	**0.549**	**0.518**
Mead	0.489	0.499	0.270	0.276	0.477	0.488
SumBasic	0.423	0.453	0.207	0.219	0.408	0.437

Table 6 Summary generated by the LUHN algorithm (having highest ROUGE score), for the Hyderabad blast dataset

RT @krajesh4u: Reports of explosion from busy commercial area in Hyderabad [url]
RT @SRIRAMChannel: Bomb blast in dilsukhnagar (hyderabad) near venkatadri theatre.many feared dead
RT @abpnewstv: BREAKING: 7 feared dead in Hyderabad blast - Reports
RT @ndtv: Bomb blast in Hyderabad: 50 injured, say officials
RT @abpnewstv: BREAKING: Twin blast in Hyderabad's Dilsukh Nagar suburb - reports of 15 deaths, over 50 injured
RT @BreakingNews: 2 blasts reported near bus stand in southern Indian city of Hyderabad; 10 people feared dead, at least 40 others injur
RT @abpnewstv: BREAKING: 9 killed, 32 injured in serial blasts in Hyderabad: PTI quoting official sources
RT @ibnlive: #Hyderabadblasts: High alert declared across Andhra Pradesh #IBNnews
RT @IndianExpress: FLASH: 9 killed, 32 injured in serial powerful #blast in Dilshuknagar area in #Hyderabad: Police
[url] wrote: Hyderabad blast: High alert declared across Andhra Pradesh
RT @SkyNewsBreak: UPDATE: AFP - police say seven people have died and 47 people hurt in bomb blasts in Indian city of #Hyderabad
'9 killed in Hyderabad blast; 5 in police firing' [url] #BengalRiots #HyderabadBlast #HinduGenocide
RT @ndtv: Hyderabad serial blasts: Mumbai, Karnataka put on high alert
RT @IndiaToday: 7 feared dead, 20 others injured in 5 blasts in Hyderabad: Report: The blasts took place in a cro
RT @ndtv: Hyderabad serial blasts: at least 15 dead, 50 injured [url]
RT @SkyNews: Hyderabad Blast: 'Multiple Deaths' [url]
#india #business : Seven killed in Hyderabad blast, several injured: Times Now: Seven killed in Hyderabad blast
"@SkyNews: #Hyderabad Blast: 7 Feared Dead [url]" what's happening now
RT @bijugovind: Screen map of #Hyderabad blast are" [url]
Seven killed in Hyderabad blast, 7 feared dead: Times Now: Seven killed in Hyderabad blast, 7 feared dead: Tim
Blasts rocked Hyderabad city many Killed & Injured [url] #Hyderabad #Blast #Dilsukh Nagar #Police. RT @SaradhiTweets: list of hospitals in hyderabad [url] #hyderabadblasts
Bomb blast in Hyderabad in busy commercial area [url]
RT @dunyanetwork: (Breaking News) Twin blasts in #Hyderabad, #India 7 people killed, 20 injured Casualties expected to rise
Explosions Rock Hyderabad; At Least 20 Killed #Hyderabadblast #hyderabad blast [url]

the summary generated by LexRank is dominated by only one type of information—regarding casualties—in which the summary generated by LUHN has much more diverse information.

It should also be noted that the best performing algorithm achieves a ROUGE-1 Recall score of 0.563, and ROUGE-2 Recall score of 0.331, which roughly implies

Table 7 Summary generated by the LexRank algorithm (having lowest ROUGE score), for the Hyderabad blast dataset

RT @krajesh4u: Reports of explosion from busy commercial area in Hyderabad [url]
RT @SRIRAMChannel: Bomb blast in dilsukhnagar (hyderabad) near venkatadri theatre.many feared dead.
RT @Iamtssudhir: Explosion took place near venkatdri theatre in dilsukhnagar
RT @abpnewstv: BREAKING: 7 feared dead in Hyderabad blast - Reports
RT @ndtv: Bomb blast in Hyderabad: 50 injured, say officials.
#Hyderabad blast took place around 7 p.m. local time; not believed to be gas cylinder explosion, @timesnow reporting
RT @khaleejtimes: Breaking News: Seven killed in Hyderabad blast [url]
Hyderabad Blast.
RT @EconomicTimes: #Hyderabad blast: Seven killed, several injured [url]
15 killed 50 injured in Hyderabad blast More Photos: [url] [url]
'9 killed in Hyderabad blast; 5 in police firing' [url] #BengalRiots #HyderabadBlast #HinduGenocide
RT @IndiaToday: 7 feared dead, 20 others injured in 5 blasts in Hyderabad: Report: The blasts took place in a cro...
RT @ndtv: Hyderabad serial blasts: at least 15 dead, 50 injured [url]
Blasts at Hyderabad [url] #News
Screen map of Dilsukhnagar #Hyderabad blast [url]
RT @ndtv: Alert in all major cities across India after serial blasts in Hyderabad.
Three blasts in Hyderabad.Fuck.
Reports of Blasts from Hyderabad @ndtv
Blasts rocked Hyderabad city many Killed & Injured [url] #Hyderabad #Blast #Dilsukh Nagar #Police
RT @timesofindia: Hyderabad Police: Two bomb blasts
11 people were killed and 50
RT @anupamthapa: Seven feared killed, 20 injured in Hyderabad blast
TimesNow : 7 feared Killed 20 injured #Hyderabadblasts thats very horryfying news
Bomb blasts near Dilsukhnagar bus stand in Hyderabad; at least 7 people injured
Explosions Rock Hyderabad; At Least 20 Killed #Hyderabadblast #hyderabad blast [url]

that the algorithmic summaries can capture only about half of the unigrams and 33% of the bigrams in the gold standard summaries. These moderate ROUGE values reiterate that summarization of microblogs posted during emergency events is a challenging problem, for which improved algorithms need to be developed in future.

6 Conclusion

Summarization of microblogs posted during emergency situations is an important and practical problem. While a large number of summarization algorithms have been proposed in the literature, to our knowledge, there has not been any systematic comparison of how effective different algorithms are in summarizing microblogs related to disaster events. In this work, we perform such a comparison of eight extractive summarization algorithms, over microblogs posted during five disaster events. We find that different algorithms generate vastly different summaries, and while some algorithms (e.g. LUHN, COWTS) achieve relatively high ROUGE scores, some other algorithms such as LexRank do not appear so effective.

We believe that the present work indicates several research directions for the future. First, given that even the best performing methods achieve ROUGE recall scores of less than 0.6, it is evident that better algorithms are needed for effectively summarizing microblogs during disaster events. Second, since different summarization algorithms produce very different summaries from the same input data, a promising direction can be to investigate whether outputs from multiple summarization algorithms can be combined to produce summaries that are better than those produced by the individual algorithms. We plan to pursue these directions in future.

References

1. Imran, M., Castillo, C., Diaz, F., Vieweg, S.: Processing social media messages in mass emergency: a survey. ACM Comput. Surv. **47**(4), 67:1–67:38 (2015)
2. Das, D., Martins, A.F.: A survey on automatic text summarization. Lit. Surv. Lang. Stat. II Course CMU **4**, 192–195 (2007)
3. Gupta, V., Lehal, G.S.: A survey of text summarization extractive techniques. IEEE J. Emerg. Technol. Web Intell. **2**(3), 258–268 (2010)
4. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**(2), 159–165 (1958)
5. Rudra, K., Ghosh, S., Goyal, P., Ganguly, N., Ghosh, S.: Extracting situational information from microblogs during disaster events: a classification-summarization approach. In: Proceedings of ACM CIKM (2015)
6. Olariu, A.: Efficient online summarization of microblogging streams. In: Proceedings of EACL(short paper), pp. 236–240 (2014)
7. Shou, L., Wang, Z., Chen, K., Chen, G.: Sumblr: continuous summarization of evolving tweet streams. In: Proceedings of ACM SIGIR (2013)
8. Wang, Z., Shou, L., Chen, K., Chen, G., Mehrotra, S.: On summarization and timeline generation for evolutionary tweet streams. IEEE Trans. Knowl. Data Eng. **27**, 1301–1314 (2015)
9. Zubiaga, A., Spina, D., Amigo, E., Gonzalo, J.: Towards real-time summarization of scheduled events from twitter streams. In: Hypertext(Poster) (2012)
10. Erkan, G., Radev, D.R.: LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization, pp. 457–479 (2004)
11. Dutta, S., Ghatak, S., Roy, M., Ghosh, S., Das, A.K.: A graph based clustering technique for tweet summarization. In: 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), pp. 1–6. IEEE (2015)
12. Xu, W., Grishman, R., Meyers, A., Ritter, A.: A preliminary study of tweet summarization using information extraction. In: Proceedings of NAACL **2013**, 20 (2013)
13. Chakrabarti, D., Punera, K.: Event summarization using tweets. In: Proceedings of AAAI ICWSM, pp. 340–348 (2011)
14. Gillani, M., Ilyas, M.U., Saleh, S., Alowibdi, J.S., Aljohani, N., Alotaibi, F.S.: Post summarization of microblogs of sporting events. In: Proceedings of International Conference on World Wide Web (WWW) Companion, pp. 59–68 (2017)
15. Khan, M.A.H., Bollegala, D., Liu, G., Sezaki, K.: Multi-tweet summarization of real-time events. In: Socialcom (2013)

16. Nichols, J., Mahmud, J., Drews, C.: Summarizing sporting events using twitter. In: Proceedings of ACM International Conference on Intelligent User Interfaces (IUI), pp. 189–198 (2012)
17. Takamura, H., Yokono, H., Okumura, M.: Summarizing a document stream. In: Proceedings of ECIR (2011)
18. Osborne, M., Moran, S., McCreadie, R., Lunen, A.V., Sykora, M., Cano, E., Ireson, N., Macdonald, C., Ounis, I., He, Y., Jackson, T., Ciravegna, F., OBrien, A.: Real-time detection, tracking, and monitoring of automatically discovered events in social media. In: Proceedings of ACL (2014)
19. Kedzie, C., McKeown, K., Diaz, F.: Predicting salient updates for disaster summarization. In: Proceedings of ACL (2015)
20. Nguyen, M.T., Kitamoto, A., Nguyen, T.T.: Tsum4act: a framework for retrieving and summarizing actionable tweets during a disaster for reaction. In: Proceedings of PAKDD (2015)
21. Inouye, D.I., Kalita, J.K.: Comparing twitter summarization algorithms for multiple post summaries. In: Proceedings of IEEE SocialCom/PASSAT, pp. 298–306 (2011)
22. Mackie, S., McCreadie, R., Macdonald, C., Ounis, I.: Comparing algorithms for microblog summarisation. In: Proceedings of CLEF (2014)
23. Rosa, K.D., Shah, R., Lin, B., Gershman, A., Frederking, R.: Topical Clustering of Tweets
24. Garg, N., Favre, B., Riedhammer, K., Hakkani-Tr, D.: Clusterrank: a graph based method for meeting summarization. In: INTERSPEECH, pp. 1499–1502. ISCA (2009)
25. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. J. Artif. Int. Res. **22**(1), 457–479 (2004)
26. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: SIGIR, pp. 19–25 (2001)
27. Ozsoy, M.G., Alpaslan, F.N., Cicekli, I.: Text summarization using latent semantic analysis. J. Inf. Sci. **37**(4), 405–417 (2011). http://dx.doi.org/10.1177/0165551511408848
28. Radev, D.R., Allison, T., Blair-Goldensohn, S., Blitzer, J., elebi, A., Dimitrov, S., Drbek, E., Hakim, A., Lam, W., Liu, D., Otterbacher, J., Qi, H., Saggion, H., Teufel, S., Topper, M., Winkel, A., Zhang, Z.: MEAD—a platform for multidocument multilingual text summarization. In: LREC. European Language Resources Association (2004)
29. Radev, D.R., Hovy, E., McKeown, K.: Introduction to the special issue on summarization. Comput. Linguist. **28**(4), 399–408 (2002)
30. Nenkova, A., Vanderwende, L.: The impact of frequency on summarization. Technical report, Microsoft Research (2005)
31. Hyderabad blasts—Wikipedia (2013). http://en.wikipedia.org/wiki/2013_Hyderabad_blasts
32. Sandy Hook Elementary School shooting–Wikipedia (2012). http://en.wikipedia.org/wiki/Sandy_Hook_Elementary_School_shooting
33. North India floods—Wikipedia (2013). http://en.wikipedia.org/wiki/2013_North_India_floods
34. Typhoon Hagupit—Wikipedia (2014). http://en.wikipedia.org/wiki/Typhoon_Hagupit
35. 2015 Nepal earthquake—Wikipedia (2015). http://en.wikipedia.org/wiki/2015_Nepal_earthquake
36. REST API Resources, Twitter Developers. https://dev.twitter.com/docs/api
37. Tao, K., Abel, F., Hauff, C., Houben, G.J., Gadiraju, U.: Groundhog day: near-duplicate detection on twitter. In: Proceedings of Conference on World Wide Web (WWW) (2013)
38. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Proceedings of Workshop on Text Summarization Branches Out, ACL, pp. 74–81 (2004)