Mining Graph-based Features in Multi-objective Framework for Microblog Summarization

Naveen Saini*, Sushil Kumar, Sriparna Saha, Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology Patna, Bihar, India-801106
Email: {naveen.pcs16, 1811cs17, sriparna, pb}@iitp.ac.in

***Abstract*—Nowadays, micro-blogging sites are getting popular due to the involvement of a large number of users. In the case of natural disasters, a significant amount of relevant information (giving crucial information) are present amongst the tweets. Therefore, there is a need to develop a system that summarizes relevant tweets by extracting informative tweets. In the current paper, we have proposed an unsupervised approach for summarizing the relevant tweets namely, *MOOTweetSumm+*, which automatically selects the informative tweets. Several tweet-scoring measures: (a) anti-redundancy measuring the dissimilarity between tweets; (b) similarity with outputs provided by LexRank (a graph-based method measuring tweet importance based on the concept of eigen-vector centrality in a graph); (c) BM25 based ranking function; (d) tf-idf based ranking function; (e) length of the tweet; (f) re-tweet count, are simultaneously optimized utilizing a binary differential evolution algorithm. Further, two different versions of the LexRank, utilizing syntactic and semantic similarity, have also been explored. For evaluation, four different disaster-event related datasets are used, and performance is measured in terms of ROUGE scores. An ablation study is also performed to determine which set of measures is best suited for different datasets. From the results obtained, it is clearly evident that our approach improves by 13.2% and 5.8% in terms of ROUGE-2 and ROUGE-L scores, over the existing approaches, respectively.**

***Index Terms*—Microblog, disaster-event, extractive summarization, multi-objective optimization, evolutionary algorithm, LexRank.**

I. Introduction

Due to continuous growth in the social media platforms like Twitter, Tumblr[1], etc., a lot of short-text massages called as tweets, are posted related to various categories like education, political issues, disaster-event, etc. and thus, have become the invaluable source of getting updated information of ongoing events [1]. As per the Twitter blog[2] posted in 2013, 400 million tweets are created by the 200 active users per day. In 2016 and 2019, this number is increased to 303 million and 500 million tweets per day[3]. This proves the vast amount of increasing information day-by-day. These tweets are posted with varying characteristics in terms of relevancy (providing useful information) or non-relevancy. This makes the relevant tweet or information extraction from such data a crucial task. But, if such relevant information gets extracted fruitfully, then it may help in the decision-making process. Another challenge is to deal with the extracted relevant tweets because going through all such tweets is a time-consuming task - this demands summarization of the relevant tweets [2], [3].

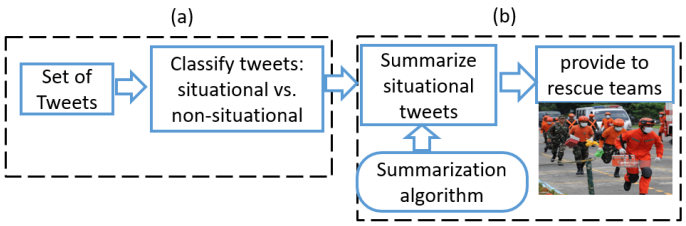

Fig. 1. Figure showing (a) classification of tweets into situational and non-situational categories; (b) summarization of situational tweets.

In this paper, we have considered disaster event-related tweets as summarizing those may help the Govt bodies or rescue teams in managing the situation of the affected area. Here, let us call relevant and non-relevant tweets as situational and non-situational tweets, respectively. Situational tweets [2] include those tweets which provide information about the current situation of the affected area, number of casualties or some other crucial information. While, non-situational tweets are related to sympathy, emotions, post disaster-event analysis. In Figure 1, the general flow of classification vs. summarization is shown. The focus of this paper is on part (b) of Figure-1.

A. Related Works

In the literature, there exist a lot of works on microblog summarization. Some of the well-known existing summarization algorithms are cluster-rank [4], LUHN [5], LSA [6], MEAD [7], LexRank [8], SumDSDR [9], SumBasic [10], FreqSum [10]. The paper by Dutta et al. [11] shows the comparison of these algorithms after application on disaster-related tweets. Some recent methods include COWTS [2], EnGraphSumm [12] and MOOTweetSumm [13]. Brief descriptions of some of the algorithms are shown in Table I.

B. Objective of the Paper

Nowadays, a lot of works have been conducted on graph-based summarization [16], [17] to summarize single document [18], multiple documents [19], biomedical article, microblogs [13], etc. Some examples of unsupervised graph-based summarization algorithms are TextRank [20], LexRank [8] which

[1]https://www.tumblr.com/tagged/social-networking
[2]https://blog.twitter.com/official/en_us/a/2013/celebrating-twitter7.html
[3]https://www.dsayce.com/social-media/tweets-day/

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

TABLE I
EXISTING METHODS WITH THEIR BRIEF DESCRIPTIONS.

Method	Year	Proposed Work
LexRank [8]	2004	(a) Graph-based method; (b) Compute sentence relevance in the graph using the idea of eigen-vector centrality.
LSA [8]	2001	Utilize the concept of singular value decomposition applied on terms-by-sentences matrix.
MEAD [7]	2004	Its a centroid based method and uses the centroids of the clusters to identify the central sentences corresponding to each cluster.
CMLDA [14]	2014	This paper explores microblog summarization with multimedia (texts, videos and images) information.
COWTS [2]	2015	(a) Authors developed a classification vs. summarization framework; (b) First classifies tweets to extract situational and non situational information, and then, tweets having highest scores in terms of content words (numerals, noun, verbs) are selected to form a microblog summary.
CNN [15]	2016	(a) Uses convolution neural network (CNN) to perform opinion-based microblog summarization; (b) works on Chinese language. (c) First identifies the polarity of a microblog using CNN and then, a feature graph is constructed based on TextRank; (c) Finally, most semantically related tweets w.r.t top-ranked features are extracted.
DEPSUB [3]	2018	(a) Proposed a framework to identify the sub-events; (b) Uses the concept of Integer Linear Programming to generate summaries of a large volume of tweets.
EnGraphSumm [12]	2018	(a) Authors of this method have considered the summaries provided by the different existing summarization algorithms like LSA, LexRank, etc.; (b) then, a single summary is generated using the ensembling strategy.
MOOTweetSumm [13]	2019	(a) Utilizes the concept of multi-objective optimization for microblog summarization; (b) Considers different tweet-scoring functions like tf-idf score, length of the tweets and anti-redundancy, and simultaneously optimizes them to generate a good quality summary.

compute the sentence (tweet) importance score based on various features and select the top sentences to form a summary. For example, in LexRank, tweet importance is calculated using the concept of eigen-vector centrality in a graph and then, top scoring tweets are selected as a part of summary. The importance of using graphs in summarization can be analyzed from the recent paper [21] on neural-network based multi-document summarization in which sentence relation graph is incorporated to summarize multiple documents.

Moreover, nowadays, the concept of multi-objective evolutionary algorithm (MEA) is getting popular [22], [23] and huge improvements have been reported after their applications on real-life problems like clustering [24], summarization [13], [18], [19]. In MEA, more than one objective functions are simultaneously optimized using the evolutionary procedure like genetic algorithm [22], etc. Motivated by this, we have proposed an algorithm, *MOOTweetSumm+*, for microblog summarization, by fusing the advantages of MEA and graph-based concept. Here, our task is to select the optimal subset of tweets by simultaneously optimizing two objective functions: (a) anti-redundancy measuring the dissimilarity between the tweets; (b) graph-based feature utilizing LexRank, for computing tweet importance score (more details are provided in section II). For optimization, differential evolution (DE) algorithm is utilized which is an evolutionary algorithm. The idea behind LexRank is based on 'recommend' or in other words, if a tweet is very similar to many other tweets, then, it must have great importance. For more details about LexRank, one can refer to [8].

Two additional tweet scoring features are also considered in our optimization framework: (a) BM25 [12], a bag-of-word based retrieval function designed to rank the short-texts; (b) RT (re-tweet) [14] which counts how many times a tweet is re-posted. A high value of re-post/re-tweet indicates that it has lot of attention and interest from the users. A significant value of RT attracts a lot of attention and is having more importance. Similar is the case with BM25. Noted that these functions are never explored in integration with MEA for microblog summarization task.

As per the survey, we have found that the existing summarization systems [8], [11], [12] utilizing LexRank algorithm, making use of syntactic similarity for computing similarity between sentences/tweets. But, in the current work, we have explored the LexRank utilizing semantic similarity and to measure the same, it utilizes word mover distance (WMD) [25] as a similarity measure. Note that WMD makes use of pre-trained word2vec [4] tool [26] which contains word vectors of several hundred dimensions and trained on 53 million tweets.

It is important to note that *MOOTweetSumm+* is the extension of our preliminary/baseline version developed for microblog summarization task namely, *MOOTweetSumm*. In *MOOTweetSumm*, the concept of *multi-objective optimization (MOO)* is utilized which considers various statistical measures and simultaneously optimizes them to optimize the quality of the summary. These measures are (a) anti-redundancy measuring the dissimilarity between the tweets; (b) tf-idf score of the tweet (sum of tf-idf scores of the words in the tweet); (c) length of the tweet, and all these objective functions are of maximization type. At the end, it provides a set of Pareto optimal solutions equally important to each other and user can select any solution based on his/her choice. It also utilizes DE algorithm as the underlying optimization strategy.

C. Contributions

The major contributions of this paper are as follows: (a) Graph-based feature has never been explored in integration with multi-objective optimization for solving microblog summarization task. Therefore, we have developed a multiobjective based microblog summarization system utilizing graph based features extracted using LexRank algorithm in providing good quality summary; (b) The existing LexRank algorithm uses syntactic similarity to measure the similarity amongst

[4]https://crisisnlp.qcri.org/lrec2016/lrec2016.html

TABLE II
NOTATIONS USED WITH THEIR DESCRIPTIONS. HERE, TF-IDF REFERS TO *Term frequency-inverse document frequency*.

Symbol	Description
$\mathcal{E}$	Disaster event containing tweets
$\mathcal{N}_E$	Total number of situational tweets in $\mathcal{E}$
$\mathcal{M}$	Number of tweets to be in the summary
$\mathcal{S}$	Obtained Summary
t_k	kth tweet
t_{avg}	Average number of words per tweet
$\mid t_k \mid$	Number of words in the kth tweet
$\mathcal{T}(w, t_k)$	Term frequency of a word 'w' in kth tweet
$\mathcal{F}(w, t_k)$	Inverse-document frequency of a word 'w' in kth tweet
$\mathcal{I}(w, t_k)$	tf-idf ($\mathcal{T}(w, t_k) \times \mathcal{I}(w, t_k)$) score of a word 'w' in kth tweet
$\mathcal{L}(t_k)$	Length of kth tweet (t_k)
$\mathcal{D}(t_k, t_m)$	Word move distance between kth and mth tweets.
$\mathbb{P}$	Population
$\mid \mathbb{P} \mid$	Number of solutions in the population
$MaxGen$	Maximum number of generations
CR	Crossover probability
F	Control factor
b	Real positive constant

tweets. But, here, we have also explored the impact of semantic similarity measure in LexRank which considers tweets in the semantic space. To capture the semantics, deep-learning based tool *word mover distance* is utilized; (c) In the literature, the impact of BM25 and re-tweet scoring functions are well explored [12] in checking the importance of tweets. But, these objective functions were never used with MEA as the optimization criteria. Therefore, we have explored these along with other objective functions; (d) Given a data set, it is difficult to decide the optimal set of objective functions. Therefore, in the current work, ablation study has been performed on various objective function combinations to see the best candidate set of tweet-scoring functions.

For evaluation, four disaster-event related datasets are used, and performance is evaluated in terms of ROUGE measure. Results clearly illustrate that the incorporation of graph-based tweet-scoring function as one of the objectives helps in improving the quality of the summary obtained. Further, results are validated using a statistical significance t-test.

II. VARIOUS STATISTICAL MEASURES/TWEET SCORING FUNCTIONS

For any summarization system, selection of various statistical measures, helping in selection of informative sentences/tweets, is a crucial task. Therefore, in this paper, we have explored six measures (also called as objective functions or tweet-scoring functions) and all should be maximized to obtain a good quality summary. Mathematical formulations of these functions are discussed below. Notations/symbols used all over the paper are described in the Table II.

1) MaxAntiRedundancy (J1): To avoid from redundancy in the summary ($\mathcal{S}$), all the tweets in $\mathcal{S}$ should be different from each other. It is defined as

$$J1 = \left(\sum_{k,l=1,k\neq l}^{\mathcal{M}} \mathcal{D}(t_k, t_l) \right) / \mathcal{M} \tag{1}$$

2) MaxSumTFIDF (J2): It calculates the sum of tf-idf score of tweets belonging to a summary ($\mathcal{S}$). It means; first, we have to sum up the tf-idf score of each word in the tweet and then take the average of tf-idf score of each tweet.

$$J2 = \left(\sum_{k=1}^{\mathcal{M}} \sum_{w \in t_k, t_k \in S} \mathcal{I}(w, t_k) \right) / \mathcal{M} \tag{2}$$

For a word $w \in t_k$ and $t_k \in \mathcal{S}$, $\mathcal{I}(w, t_k)$ is calculated as

$$I(w, t_k) = \mathcal{T}(w, t_k). \left(1 + \log \frac{1 + \mathcal{N}_E}{1 + \{t' \in \mathcal{E} | k \in t'\}} \right) \tag{3}$$

where, $\mathcal{T}(w, t_k)$ is the number of times the word 'w' appears in kth tweet.

3) MaxLength (J3): It is assumed that longer tweets are more informative than shorted tweets [12], [13]. To consider this scenario, this objective function was designed and can be formulated as

$$J3 = \sum_{k=1}^{\mathcal{M}} L(t_k) \tag{4}$$

Note that it was not average over the number of tweets in the summary. The reason is described using an example: Suppose summary A has 20 tweets and summary B has 21 tweets including the 20 tweets that are also in A. The additional tweet has a length of 1 (1 word), then the average length of summary B will be smaller than that of A which is contradictory to our thinking.

4) MaxOverlapLexRank (J4/J5): It counts the number of overlapping tweets with the top-scoring tweets identified by the LexRank [8] algorithm utilizing two different similarity measures: syntactic and semantic. First one makes use of tf-idf vector representation of the tweets and then, calculates the cosine similarity to find the relatedness between tweets. While, second one makes use of word mover distance to evaluate the dissimilarity between tweets. Let L1 and L2 be the top scoring tweets identified using syntactic and semantic similarity, respectively; then score of this function can be obtained as

$$J4 = \mid S1 \cap L1 \mid \quad \text{and} \quad J5 = \mid S1 \cap L2 \mid \tag{5}$$

where, S1 is the set of sentences in the summary $\mathcal{S}$.

5) MaxSumBM25 (J6): BM25 [27] is the ranking function in information retrieval used to rank the documents (tweets in our case) based on relevance to the query. It was basically designed for short texts like tweets. As per literature [12], it performs better than tf-idf [28] model when text is short-length like tweets. Therefore, it is adopted as one of the objective functions in our framework.

$$J6 = \Bigg(\sum_{k=1, t_k \in S}^{\mathcal{M}} \underset{t_k \in S}{BM25}(t_k, Q) \Bigg) / \mathcal{M} \tag{6}$$

where, Q is a query with terms $q_1, q_2, ...q_n$ and BM25 score of a tweet $t_k \in S$ denoted as $\mathcal{W}$ (=$BM25(t_k, Q)$) is defined as

$$\mathcal{W} = \sum_{i=1}^{n} \mathcal{F}(q_i, t_k) \frac{\mathcal{T}(q_i, t_k) \cdot (k_1 + 1)}{\mathcal{F}(q_i, t_k) + k_1 \cdot (1 - b + b \cdot (|t_k|/t_{avg}))} \tag{7}$$

where, b and k_1 are hyper parameters for BM25. Note that here Q refers to the entire set of tweets in the disaster event $\mathcal{E}$.

6) MaxRTScore (J7): On any social network, importance of the tweet can be revealed from the re-post number [14]. A high value of re-post indicates that it has lot of attention and interest from the users. Therefore, to evaluate the quality of summary $\mathcal{S}$, it is evaluated as

$$J7 = \sum_{k=1}^{\mathcal{M}} \log(RepostNumber(t_k) + 1) \tag{8}$$

where, $RepostNumber$ counts how many times a tweet is re-posted.

Note that first three objective ($J1$, $J2$ and $J3$) functions are same as discussed in our preliminary model *MOOTweetSumm*.

III. Problem Statement

The aim of the current paper is to select the optimal (near-optimal) set of tweets by simultaneously optimizing various statistical measures discussed in Section II. If $\mathcal{E}$ is any disaster event containing $\mathcal{N}_E$ number of tweets, then our task is to obtain a summary $\mathcal{S}$, consisting of $\mathcal{M}$ number of tweets belonging to $\mathcal{E}$ then

$$\max\{J1(\mathcal{S}), J2(\mathcal{S}), J3(\mathcal{S}), J4(\mathcal{S}), J5(\mathcal{S}), J6(\mathcal{S}), J7(\mathcal{S})\} \tag{9}$$

These functions are simultaneously optimized using the multi-objective binary differential evolution (MBDE) [23] algorithm, which is a population-based meta-heuristic algorithm. Here, the population consists of a set of solutions represented in the form of binary vectors, and each solution is associated with fitness/objectives values. At the end the algorithm, it provided a set of Pareto optimal solutions out of which the best solutions is selected based on user choice.

Note that (a) we have performed the ablation study by varying the objective function combinations; for example, {J1, J2}, {J1, J2, J3}, {J1, J5, J6}, {J1, J6, J7} are some possible sets of objective functions which need to be simultaneously optimized in different runs of the proposed algorithm (we have tried up to maximum 3 objective functions); (b) $J1$ is kept common to cover diverse set of tweets.

Algorithm 1 Procedure of MOOTweetSumm+

1: $\mathbb{P} \leftarrow$ Initialize Population $< X^1, X^2, X^3, \ldots, X^{|\mathbb{P}|} >$
2: For each solution $\mathcal{X}$, evaluate objective functional values
3: $CGen$=0 ▷ Current generation number
4: Repeat step-5 to 9 until $CGen < MaxGen$
5: $\mathbb{P}^{'}$=[] ▷ Population to store new solutions
6: For each solution $\mathcal{X} \in \mathbb{P}$, generate new solution
   (a) Randomly select three solutions $r1, r2$ and $r3$ from $\mathbb{P}$ to form a mating pool
   (b) $Prob(\mathcal{X}) \leftarrow$ Perform probability estimation operator using selected random solutions and $\mathcal{X}$
   (c) $Y^{'} \leftarrow$ Convert $Prob(\mathcal{X})$ into a binary solution
   (d) $Y^{''} \leftarrow$ Perform crossover between $Y^{'}$ and $\mathcal{X}$
   (e) Evaluate objective functions for $Y^{''}$
   (f) Add $Y^{''}$ into $\mathbb{P}^{'}$
7: Merge Old population ($\mathbb{P}$) and new population ($\mathbb{P}^{'}$)
8: $\mathbb{P} \leftarrow$ Select the best $|\ \mathbb{P}\ |$ solutions based on their objective functional values using non-dominating sorting and crowding distance operator
9: $CGen \leftarrow CGen$+1
10: **return** the best summary

IV. Proposed Method

In the current paper, the steps of our approach (MOOTweetSumm+) are shown in Algorithm 1. As our algorithm is based on differential evolution (DE) framework, therefore, it starts from a set of some random binary solutions, called as population (step-1). The length of these solutions is kept equal to the number of tweets in the dataset. Note that the number of 1's in each solution should not exceed $\mathcal{M}$.

Then the objective functions which need to be simultaneously optimized are evaluated for each solution (step-2). Afterward, the iterative procedure begins (step-5 to step-9) starting from $0th$ generation and continues until the maximum number of generations is reached. In step-6, a new solution (also called a trail solution in DE) generation takes place for each solution in the population. Various genetic operators like mating pool construction (step-6(a)), mutation (step-6(b) and 6(c)) and crossover (step-6(d)), are applied in forming a new solution.

For mutation, firstly, a probability estimation operator is performed between chosen random solutions in step-6(a) and current solution '$\mathcal{X}'$ as follows:

$$Prob(\mathcal{X}_j) = \frac{1}{1 + e^{-\frac{2b \times [\mathcal{X}_{r1,j} + F \times (\mathcal{X}_{r2,j} - \mathcal{X}_{r3,j}) - 0.5]}{1 + 2F}}} \tag{10}$$

where, $\mathcal{X}_j$ and $\mathcal{X}_{r,j}$ denote the jth component of the current solution '$\mathcal{X}'$ and chosen random solution(s) 'r' (r1/r2/r3), $\mathcal{F}$ and b are the DE scaling/weight factor and real positive constant, $(\mathcal{X}_{r1,j} + F \times (\mathcal{X}_{r2,j} - \mathcal{X}_{r3,j}) - 0.5)$ is the mutation

operation. Eq. 10 provides probability values for different components of the current solution and then, based on some heuristic, those are converted into binary values as shown below:

$$Y'_j = \begin{cases} 1, \text{ if rand()} \leq Prob(\mathcal{X}_j) \\ 0, otherwise \end{cases} \quad (11)$$

Above equation gives rise a solution Y'. Then, crossover is performed between Y' and he current solution $\mathcal{X}$ as

$$Y''_j = \begin{cases} Y'_j, \text{if rand()} \leq CR \\ \mathcal{X}_j, \text{Otherwise} \end{cases} \quad (12)$$

where, Y'' is a new solution generated for the current solution $\mathcal{X}$ and CR is the crossover probability. Similarly, for all the solutions, new solutions are generated, and objective functions are evaluated. If $\mid \mathbb{P} \mid$ is the number of solutions in the population, then the equal number of new solutions are generated, thus on merging, total $\mid 2\mathbb{P} \mid$ solutions will be there, out of which best $\mid \mathbb{P} \mid$ solutions are selected after application of non-dominated sorting and crowding distance operator [22]. In the final generation, we will get a set of Pareto optimal solutions out of which single best solution is selected, having a good summary. For more detail, reader can refer to the baseline paper [13].

V. Experimental Setup

A. Datasets Used

For our experimentation, we have used four datasets related to different disaster events, which are (a) Bomb blasts in Hyderabad, India; (b) Flood in Uttarakhand state, India; (c) Sandyhook elementary school shooting, USA; (d) Typhoon hagupit in the Philippines. The number of tweets available in these datasets are 1413, 2069, 2080, and 1461, respectively. Note that all the tweets in these datasets are situational tweets. Same datasets are used by the papers [12], [13]. The actual/gold/reference summary is also provided with these datasets. The number of tweets available in the actual summary corresponding to these datasets are 33, 34, 37, and 41, respectively. More details about these datasets can be found in [13].

B. Parameters Used

The proposed approach (*MOOTweetSumm+*) is the extension of the multiobjective based tweet summarization technique, *MOOTweetSumm* developed in the paper [13]. Therefore, same set of parameters as used in *MOOTweetSumm* are utilized in our framework. The values of these parameters namely, Population Size ($\mid \mathcal{P} \mid$), maximum number of generations ($MaxGen$), crossover probability (CR), F, and b are 25, 25, 0.8, 0.8, and 6, respectively. For *MaxOverlapLexRank* tweet-scoring function, top-scoring 70 tweets are considered for $L1$ and $L2$, obtained using syntactic and semantic similarity, respectively (discussed in section II). To evaluate the WMD between two tweets, pre-trained word2vec [26] model[5] trained on 53 million tweets related to various disaster events, is utilized. To calculate BM25 score of each tweet, we have utilized the code with default parameters available at the Github repository[6]. The results reported are the average over 5 runs of the proposed algorithm.

C. Evaluation Measure

To evaluate the performance of our generated/predicted summary with respect to the gold summary, ROUGE-N score is used, which is a well-known measure in any summarization system. It measures the overlapping units between generated and gold summary. In our case, N takes the value of 1, 2, and 3 to provide ROUGE-1, ROUGE-2, and ROUGE-L, respectively. For a good quality summary, the higher value of ROUGE score is desired. For mathematical definition of ROUGE-N, reader can refer to [13].

D. Comparative Methods

For comparison, we have considered two recent approaches developed in the year 2018 and 2019, namely, *EnGraphSumm* [12] and *MOOTweetSumm* [13]. Both approaches are totally unsupervised in nature and briefly described in Table I. Note that the second approach is our baseline approach. In *EnGraphSumm*, many versions are developed out of which we only consider top 4 versions namely, VecSim-ConComp-maxSumTFIDF, VecSim-ConComp-MaxDeg, VecSim-Community-maxSumTFIDF and VecSim-ConComp-MaxLen. Each one first generates summary using different existing algorithms and then, uses the ensembling strategy to select the tweets. These tweets are grouped by some graph-based method [12] and then from each group, one tweet is selected based on various features like maximum length of tweets, maximum degree of a node, etc. as a part of summary. Along these approaches, some other approaches like COWTS [2], Lex-Rank [8], LSA [29], LUHN [5], SumBasic [10], MEAD [7] SumDSR [9], are also taken into account for comparison.

VI. Discussion of Results

In this section, we will discuss the results obtained by our proposed approach in comparison with existing approaches followed by statistical significance test. As selecting the optimal set of objective functions for any task is a challenging problem, therefore, we have performed the ablation study using various combinations (minimum two and maximum three) of objective functions. Note that MaxAntiRedundancy is kept common to avoid from redundancy in the summary.

A. Comparison between Our Proposed Approach (MOOTweetSumm+) and Baseline Approach (MOOTweetSumm)

The results attained by *MOOTweetSumm* and *MOOTweetSumm+* are shown in Table III(a) and III(b). The best results

[5]http://crisisnlp.qcri.org/lrec2016/lrec2016.html

[6]http://ethen8181.github.io/machine-learning/search/bm25_intro.html

TABLE III

COMPARISON BETWEEN OUR BASELINE APPROACH(MOOTWEETSUMM) AND OUR PROPOSED APPROACH(MOOTWEETSUMM+). BOTH TABLES SHOW THE ROUGE SCORES ON THE DIFFERENT DATASETS USING VARIOUS COMBINATIONS OF OBJECTIVE FUNCTIONS.

	Datasets	HBlast			Sandyhook			Hagupit			UKflood		
Abb.	Objectives	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
C1	J1, J2	0.5371	0.3914	0.5371	0.5842	0.3721	0.5842	0.3845	0.2184	0.3782	0.4541	0.2822	0.4447
C2	J1, J3	0.5371	0.3931	0.5371	0.6139	0.3975	0.6073	0.3634	0.2241	0.3550	0.4471	0.2623	0.4400
C3	J1, J2, J3	0.5025	0.3534	0.5025	0.5940	0.3612	0.5874	0.3697	0.2213	0.3655	0.4494	0.2577	0.4424

(a) Results obtained by our baseline approach, MOOTweetSumm

	Datasets	HBlast			Sandyhook			Hagupit			UKflood		
Abb.	Objectives	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
C4	J1, J4	0.4975	0.3276	0.4975	0.5875	0.3739	0.5875	0.4097	0.2434	0.4034	0.4400	0.2623	0.4282
C5	J1, J5	0.4950	0.3500	0.4926	0.6007	0.4011	0.5974	0.4328	0.2766	0.4244	0.4494	0.2638	0.4447
C6	J1, J6	0.5050	0.3517	0.5050	0.5743	0.3485	0.5578	0.4013	0.2545	0.3866	0.4729	0.3221	0.4682
C7	J1, J7	0.4505	0.3155	0.4480	0.5479	0.3466	0.5380	0.3887	0.2490	0.3887	0.3906	0.2285	0.3859
C8	J1, J2, J4	0.5347	0.3862	0.5322	0.5710	0.3612	0.5611	0.3887	0.2353	0.3803	0.5059	0.3466	0.4965
C9	J1, J3, J4	0.5248	0.3793	0.5223	**0.6139**	**0.4265**	**0.6090**	0.4244	0.2642	0.4202	**0.5106**	**0.3558**	**0.5059**
C10	J1, J2, J5	0.5322	0.3862	0.5322	0.5578	0.3466	0.5479	**0.4559**	**0.3029**	**0.4517**	0.5059	0.3543	0.4965
C11	J1, J3, J5	**0.5520**	**0.3966**	**0.5520**	0.5941	0.3975	0.5809	0.3929	0.2503	0.3824	0.4353	0.2485	0.4306
C12	J1, J4, J6	0.4802	0.3414	0.4777	0.5281	0.3067	0.5149	0.3487	0.1784	0.3403	0.4071	0.2423	0.3906
C13	J1 , J5, J6	0.3936	0.2069	0.3911	0.4950	0.2740	0.4785	0.3403	0.1798	0.3193	0.3929	0.2377	0.3859
C14	J1, J4, J7	0.3985	0.2414	0.3911	0.4884	0.2668	0.4785	0.3466	0.1687	0.3298	0.3812	0.1840	0.3741
C15	J1, J5, J7	0.3886	0.2259	0.3837	0.4686	0.2523	0.4587	0.4202	0.2669	0.4118	0.4188	0.2362	0.4071

(b) Results obtained by our proposed approach, MOOTweetSumm+

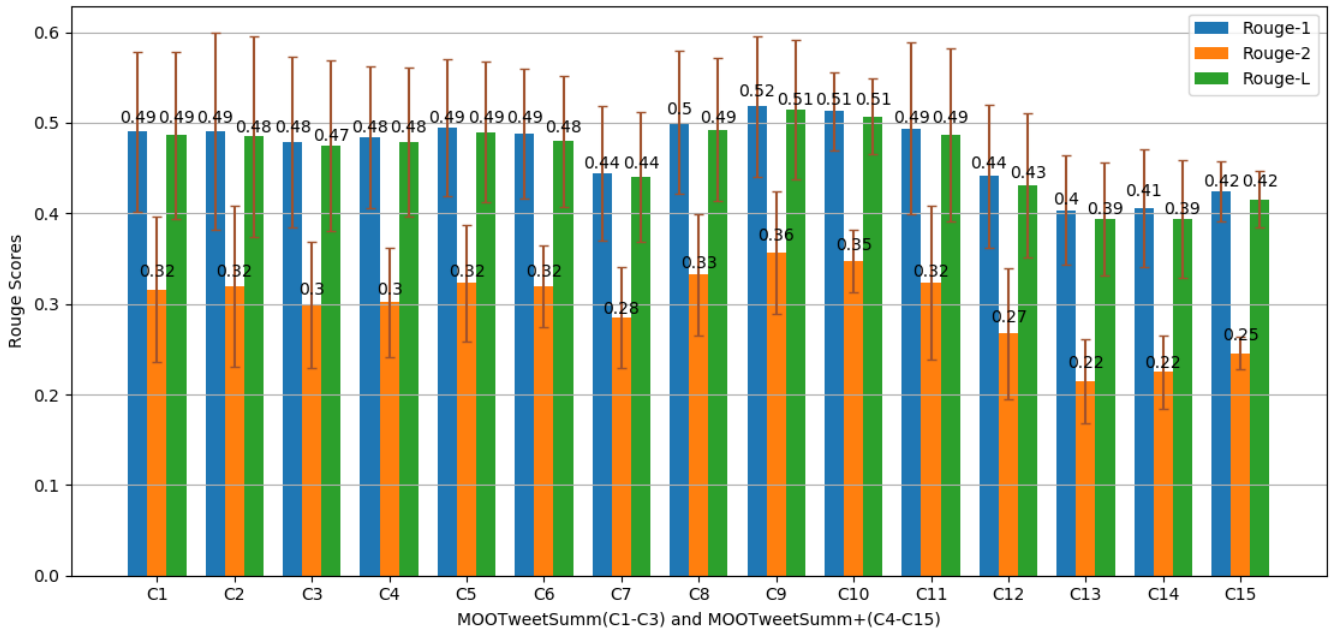

Fig. 2. Comparison of MOOTweetSumm and MOOTweetSumm+ in terms of average ROUGE scores, using different sets of objectives functions. Here, bars show the standard deviation.

among these two approaches are highlighted in bold. From these tables, it is clearly evident that graph-based objective function (J4/J5) is playing a major role when used with other objective functions. Below is the description of results over individual datasets:

1) HBlast: For this dataset, simultaneous optimization of *J1*, *J3*, and *J5* objective functions, *i.e.*, MaxAntiRedundancy, MaxLength, and graph based feature (MaxOverlapLexRank) utilizing semantic similarity, respectively, provides better result using *MOOTweetSumm+* than *MOOTweetSumm*. In terms of improvements, our approach improves by 0.9% and 2.8% over the best results of *MOOTweetSumm* in terms of ROUGE-2 and ROUGE-L scores, respectively.
2) Sandyhook: For this dataset, MOOTweetSumm+ simultaneously optimizing objective functions such as *J1*, *J3*, and *J4*, *i.e.*, MaxAntiRedundancy, MaxLength, and MaxOverlapLexRank utilizing syntactic similarity,

performs better than MOOTweetSumm and is able to improve by 7.3% and 0.3% in terms of ROUGE-2 and ROUGE-L metrics.

3) Hagupit: Here, the simultaneous optimization of MaxAntiRedundancy (J1), MaxSumTFIDF (J2), and MaxOverlapLexRank (J5) utilizing semantic similarity, respectively, by our proposed approach improves by 35.2% and 19.4% over the best result of MOOTweetSumm in terms of ROUGE-2 and ROUGE-L, respectively.
4) UKflood: Similar to Sandyhook, here also, the same set of objective functions yields better results using our *MOOTweetSumm+* and improves by 26% and 13.8% in terms of ROUGE-2 and ROUGE-L, respectively, over the best result of MOOTweetSumm.

We have also shown the average ROUGE scores over all datasets using the bar-chart, as shown in Figure 2. Here, the abbreviation $C1, C2, \ldots, C15$ denotes the various objective function combination and is shown in the first column of Table III. The objective function combinations $C1, C2$ and $C3$ are explored in *MOOTweetSumm*, while, rest are utilized in *MOOTweetSumm+*. From Figure 2, it can be inferred that the abbreviation $C9$, i.e., the objective functions, MaxAntiRedundancy (J1), MaxLength (J3), and MaxOverlapLexRank (J4) are able to provide the best average ROUGE-1, ROUGE-2 and ROUGE-L scores of 0.5184, 0.3565 and 0.5143, respectively, over all datasets. On the other hand, for MOOTweetSumm, the best average ROUGE-1, ROUGE-2, and ROUGE-L scores of 0.4900, 0.3150, and 0.4860 were attained by simultaneously optimizing MaxAntiRedundancy (J1) and MaxSumTFIDF (J2) [13] objective functions. This proves that incorporating graph-based feature in our framework as one of the objective functions, helps in improving the ROUGE scores.

B. Comparison with Existing Methods

The best results attained by our proposed approach in comparison with baseline algorithm (MOOTweetSumm) and other existing methods are shown in Table IV. It is important to note that for comparison amongst existing methodologies (excluding MOOTweetSumm), only ROUGE-2 and ROUGE-L are reported as reference papers also [2], [12] reported only these measures. From the set of objective functions explored in our framework, we have shown only the best sets beating the state-of-the-art results (highlighted in bold). In these best sets of objective functions, MaxAntiRedundancy (J1) and MaxOverlapLexRank (J4/J5) functions are common. In terms of improvement, our method utilizing J1, J3, and J4 objective functions improves by 83% and 11.8% in terms of ROUGE-2 and ROUGE-L metrics, respectively, over the best ROUGE score of EnGraphSumm. Among other existing methods like LexRank, LSA, COWTS, etc., COWTS is shown to have good performance, but, in comparison to our approach, COWTS lacks behind by 99.2% and 15.5% in terms of ROUGE-2 and ROUGE-L, respectively.

TABLE IV
AVERAGE ROUGE SCORES OVER ALL DATASETS ATTAINED BY EXISTING METHODS IN COMPARISON WITH THE BEST RESULTS OBTAINED BY THE PROPOSED APPROACH. THE SYMBOL † INDICATES THAT RESULTS ARE STATISTICALLY SIGNIFICANT AT 5% SIGNIFICANT LEVEL.

Approach	Rouge-2	Rouge-L
MOOTweetSumm+ (J1, J3, J4)	**0.3565**†	**0.5143**†
MOOTweetSumm+ (J1, J2, J5)	**0.3475**	**0.5070**
MOOTweetSumm+ (J1, J2, J4)	**0.3323**	**0.4925**
MOOTweetSumm+ (J1, J5)	**0.3229**	**0.4898**
MOOTweetSumm (J1, J2)	0.3150	0.4860
VecSim–ConComp–MaxLen	0.1940	0.4506
VecSim–ConComp–MaxDeg	0.1919	0.4457
VecSim–Community–maxSumTFIDF	0.1898	0.4591
VecSim–ConComp–maxSumTFIDF	0.1886	0.4600
ClusterRank (CR)	0.0859	0.2684
COWTS (CW)	0.1790	0.4454
FreqSum (FS)	0.1473	0.3602
Lex-Rank (LR)	0.0489	0.1525
LSA (LS)	0.1599	0.4234
LUHN (LH)	0.1650	0.4015
Mead (MD)	0.1172	0.3709
SumBasic (SB)	0.1012	0.3289
SumDSDR (SM)	0.0985	0.2602

TABLE V
THE P-VALUES OBTAINED USING TABLE IV

	p-value	
Approach	ROUGE-2	ROUGE-L
MOOTweetSumm	1.094E-034	2.568E-017
VecSim–ConComp–MaxLen	7.660E-279	1.727E-073
VecSim–ConComp–MaxDeg	6.838E-283	3.733E-083
VecSim–Community–maxSumTFIDF	6.443E-287	1.599E-057
VecSim–ConComp–maxSumTFIDF	3.305E-289	6.672E-056
ClusterRank	0.00	0.00
COWTS	3.003E-307	9.358E-084
FreqSum	0.00	2.058E-262
Lex-Rank	0.00	0.00
LSA	0.00	9.994E-130
LUHN	0.00	7.159E-177
MEAD	0.00	5.807E-241
SumBasic	0.00	1.007E-321
SumDSDR	0.00	0.00

C. Statistical Significance Test

To check whether improvements obtained by our proposed approach are statistically significant or not, in comparison to the state-of-the-art results, we have also conducted the statistical significance t-test [30] at 5% significance level. This test provides p-value. Lesser p-value indicates that the results are statistically significant. Table V shows the p-value obtained utilizing Table IV. All values are less than 5% significant level and thus prove that obtained improvements are statistically significant. Note that the best result of our proposed approach is used while computing these p-values.

VII. CONCLUSION AND FUTURE WORKS

In the current work, we have proposed a multi-objective optimization (MOO) based framework for microblog summarization, *MOOTweetSumm+*, summarizing a set of relevant

tweets. The problem is treated as a binary optimization problem where the task is to select the subset of optimal tweets by simultaneously optimizing multiple objective functions like the maximum length of the tweets, BM25 score of the tweets, the re-tweet score of the tweets, etc. Due to the popularity of graph-based algorithm (LexRank) solving different tasks, same was also integrated in our MOO framework, i.e., the maximum overlap between a subset of tweets selected and top-scoring tweets provided by LexRank algorithm utilizing some similarity measure, should be high and considered as one of the objective functions. Generally, in LexRank, the syntactic similarity measure is used to measure the similarity among tweets, but here, we have explored both syntactic and semantic similarity to observe the effect on the performance of the system developed. From the results obtained, it is clearly evident that graph-based feature when optimized along with other objective functions, is able to beat the state-of-the-art results, i.e., our approach improves by 13.2% and 5.8% in terms of ROUGE-2 and ROUGE-L, respectively, over the recently developed approach, *MOOTweetSumm*.

In the future, we would like to explore the same task using a multi-view clustering approach where firstly, tweets are clustered into various groups using multiple views and then, common partitioning is found out satisfying both the views. Finally, top-scoring tweets can be extracted from each cluster to form a summary. It is also planned to make the same task parameter adaptive, where parameters are selected adaptive instead of fixing them.

ACKNOWLEDGMENT

Dr. Sriparna Saha would like to acknowledge the support of Early Career Research Award of Science and Engineering Research Board (SERB) of Department of Science and Technology India to carry out this research.

REFERENCES

[1] M. Imran, P. Mitra, and C. Castillo, "Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France: European Language Resources Association (ELRA), may 2016.

[2] K. Rudra, S. Ghosh, N. Ganguly, P. Goyal, and S. Ghosh, "Extracting situational information from microblogs during disaster events: a classification-summarization approach," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 583–592.

[3] K. Rudra, P. Goyal, N. Ganguly, P. Mitra, and M. Imran, "Identifying sub-events and summarizing disaster-related information from microblogs," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 265–274.

[4] N. Garg, B. Favre, K. Reidhammer, and D. Hakkani-Tür, "Clusterrank: a graph based method for meeting summarization," in *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[5] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of research and development*, vol. 2, no. 2, pp. 159–165, 1958.

[6] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2001, pp. 19–25.

[7] D. R. Radev, H. Jing, M. Styś, and D. Tam, "Centroid-based summarization of multiple documents," *Information Processing & Management*, vol. 40, no. 6, pp. 919–938, 2004.

[8] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of artificial intelligence research*, vol. 22, pp. 457–479, 2004.

[9] Z. He, C. Chen, J. Bu, C. Wang, L. Zhang, D. Cai, and X. He, "Document summarization based on data reconstruction." in *AAAI*, 2012.

[10] A. Nenkova and L. Vanderwende, "The impact of frequency on summarization," *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, vol. 101, 2005.

[11] S. Dutta, V. Chandra, K. Mehra, S. Ghatak, A. K. Das, and S. Ghosh, "Summarizing microblogs during emergency events: A comparison of extractive summarization algorithms," in *Emerging Technologies in Data Mining and Information Security*. Springer, 2019, pp. 859–872.

[12] S. Dutta, V. Chandra, K. Mehra, A. K. Das, T. Chakraborty, and S. Ghosh, "Ensemble algorithms for microblog summarization," *IEEE Intelligent Systems*, vol. 33, no. 3, pp. 4–14, 2018.

[13] N. Saini, S. Saha, and P. Bhattacharyya, "Multiobjective-based approach for microblog summarization," *IEEE Transactions on Computational Social Systems*, 2019.

[14] J. Bian, Y. Yang, H. Zhang, and T.-S. Chua, "Multimedia summarization for social events in microblog stream," *IEEE Transactions on multimedia*, vol. 17, no. 2, pp. 216–228, 2014.

[15] Q. Li, Z. Jin, C. Wang, and D. D. Zeng, "Mining opinion summarizations using convolutional neural networks in chinese microblogging systems," *Knowledge-Based Systems*, vol. 107, pp. 289–300, 2016.

[16] D. Parveen and M. Strube, "Integrating importance, non-redundancy and coherence in graph-based extractive summarization," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[17] H. Zhang, M. Fiszman, D. Shin, B. Wilkowski, and T. C. Rindflesch, "Clustering cliques for graph-based summarization of the biomedical research literature," *BMC bioinformatics*, vol. 14, no. 1, p. 182, 2013.

[18] N. Saini, S. Saha, D. Chakraborty, and P. Bhattacharyya, "Extractive single document summarization using binary differential evolution: Optimization of different sentence quality measures," *PloS one*, vol. 14, no. 11, 2019.

[19] N. Saini, S. Saha, A. Kumar, and P. Bhattacharyya, "Multi-document summarization using adaptive composite differential evolution," in *International Conference on Neural Information Processing*. Springer, 2019, pp. 670–678.

[20] R. Mihalcea and P. Tarau, "Graph-based ranking algorithms for text processing," Oct. 5 2010, uS Patent 7,809,548.

[21] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. Radev, "Graph-based neural multi-document summarization," *arXiv preprint arXiv:1706.06681*, 2017.

[22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[23] L. Wang, X. Fu, M. I. Menhas, and M. Fei, "A modified binary differential evolution algorithm," in *Life System Modeling and Intelligent Computing*. Springer, 2010, pp. 49–57.

[24] N. Saini, S. Saha, and P. Bhattacharyya, "Automatic scientific document clustering using self-organized multi-objective differential evolution," *Cognitive Computation*, vol. 11, no. 2, pp. 271–293, 2019.

[25] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *International Conference on Machine Learning*, 2015, pp. 957–966.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[27] S. Robertson, H. Zaragoza *et al.*, "The probabilistic relevance framework: Bm25 and beyond," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[28] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242, 2003, pp. 133–142.

[29] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[30] B. L. Welch, "The generalization ofstudent's' problem when several different population variances are involved," *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947.