

Received August 29, 2021, accepted September 27, 2021, date of publication October 14, 2021, date of current version November 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3120112

DCBRTS: A Classification-Summarization Approach for Evolving Tweet Streams in Multiobjective Optimization Framework

DIKSHA BANSAL¹, NAVEEN SAINI², (Member, IEEE),
AND SRIPARNA SAHA¹, (Senior Member, IEEE)

¹Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna 801106, India

²AI and Big Data Department, Endicott College of International Studies, Woosong University, Daejeon 300718, South Korea

Corresponding author: Naveen Saini (naveensaini@wsu.ac.kr; nsaini1988@gmail.com)


This work was supported by Woosong University Academic Research, in 2021.

ABSTRACT The emergence of social media platforms like Twitter has become a prominent communication source in disaster outbreak. NGOs, Government agencies leverage twitter's open and public features to provide immediate relief. Nevertheless, situational information gets immersed in millions of tweets with varying characteristics. Examining each tweet can be cumbersome and time-consuming. Thus, the efficient extraction of disaster-related situational tweets and getting information from all the extracted tweets is required. In the current paper, we have developed a novel framework that uses a deep learning-based classification model to separate the situational tweets from others and summarize them in real-time. Our system is a three-phase process: (a) Creating tweet clusters using a representative set of tweets from the initial set of extracted tweets using a multi-objective optimization concept; (b) When a new tweet arrives, the clusters are updated. The new tweet is classified as situational vs. non-situational. If situational, it is assigned to the closest cluster or new cluster. This assignment is based on its weighted average of syntactic and semantic distances and relevancy to the cluster; (c) Summary is formulated by extracting tweets from each cluster. The proposed approach's superior performance on four datasets related to different disaster-related events indicates the developed framework's efficiency over the state-of-the-art techniques.

INDEX TERMS Evolving tweet-stream, summarization, classification, convolution neural network, clustering, multi-objective optimization.

I. INTRODUCTION

The increasing popularity of microblogging sites such as Twitter has changed the way people think, live, and communicate [1], [2]. As per a blog¹ in 2013, 400 million tweets are posted per day, and this number has increased to 500 million in 2019.² These sites have become a live coverage of valuable information for ongoing events such as current trends, politics, education, and more. Searching a topic can provide a lot of related tweets, which can be informative but sometimes overwhelming. In case of a disaster event, monitoring informative tweets (also called as situational tweets describing the current status of the affected area like the number of casualties, important contact numbers like blood banks) may be helpful for the disaster management

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Liu .

¹<https://blog.twitter.com/2013/celebrating-twitter7>

²<https://www.dsayce.com/social-media/tweets-day/>

authorities to carry out immediate relief operation. However, in order to process incoming tweets to perform quick response operation, two challenges may arise: (a) availability of vast amount of tweets having varied characteristics including sympathy and emotions, personal opinion, among others. In the literature [2]–[4], the importance of situational tweets has already been shown and the importance of separating situational tweets from non-situational is also established; (b) rapid rate of posting such tweets: this may cause the overload problem.

An example illustrating the situational vs. non-situational tweet is given below. The situational tweet shows some important contact numbers, while non-situational shows the sentiment of a person.:

Situational Tweet: call bsnl numbers 1503, 09412024366 to find out last active location of bsnl mobiles of missing persons in uttarakhand.

Non-Situational Tweet: Shooting was there at an elementary school. I'm losing all faith in humanity.

The current paper presents a solution by developing a two-stage approach namely, *DCBRTS*, for handling continuous tweet streams. In the first stage, tweet category is identified as either situational or non-situational using a classification framework. After that, an online summarization system is applied on the situational tweets to generate the summary in real-time (RT). Our model develops a deep learning-based classification model utilizing convolution neural network (CNN) [5] that classifies whether a tweet is situational or non-situational. The classifier uses universal sentence (tweet) representation [6] to capture semantics in above two categories. On the other hand, the summarization model resolves the overload problem by summarizing the situational tweets (obtained using classification model) as going through all such tweets is a cumbersome task for the authorities. As time is critical in disaster scenario; therefore, these tasks are performed in real-time so that extracted tweets can be made available to the authorities in a timely manner.

It is important to note that developing a real-time tweet summarization (RTS) system is not an easy task as it has to be efficient (able to handle the large tweet streams) and flexible (able to provide summaries at different breakpoints.) Most of the existing works [1], [3], [4], [7]–[9] have considered a specific trait while summarizing the tweets. For example, the approach for real-time tweet summarization in the paper [3] focuses on maximizing the number of content words (numeral, noun and verbs) using integer linear programming. But, there may be different traits like maximum length of the tweets [4], tf-idf score of the tweets [10], and anti-redundancy (to remove redundant tweets from the summary), which can be considered all together to obtain a good quality summary. Note that importance of optimizing all these traits together is shown in the paper by Saini *et al.* [2]. The paper [2] uses the multi-objective optimization concept to simultaneously optimize above stated traits and uses an evolutionary algorithm (inspired by the biological phenomenon of the nature) [11] for the purpose of optimization. It selects the optimal subset of tweets and considers them as a summary. In the current study, the approach of [2] namely, *MOOTweetSumm*, is utilized for selecting the initial set of optimal tweets from a given set of situational tweets.

As our approach is based on real-time tweet summarization; therefore, the obtained set of optimal tweets is passed to our next phase of summarization model, *clustering*, which creates groups of tweets based on their similarity. The continuous tweet-stream is assigned to different clusters considering the selected tweets (obtained using *MOOTweetSumm* [2]) as the initial cluster centers. Note that (a) for assignment, the weighted average of syntactic and semantic distance is used; (b) there is some threshold on the maximum number of clusters and the maximum number of tweets in a cluster to avoid the problem of information overload; (c) a tweet is assigned to the closest cluster based on its relevancy to that cluster which is calculated using the cluster size and the distance of the tweet from the cluster center.; (d) if an incoming tweet is closest to *i*th cluster but cannot be added due

to its non-relevancy, then a separate cluster is formed; (e) if there are more than the threshold number of clusters, then two clusters can be merged based on their centers' closeness and the number of tweets in both clusters. This is done to retain maximum information where some weight is assigned to the number of tweets in both clusters along with the distance between cluster centers so that clusters having less number of tweets and maximum similarity, can be merged.

Finally, at a certain break-point provided by the user, the summary is generated. For producing the summary, firstly, clusters present at that time-stamp are ranked based on the tweet's score in the respective clusters. The cluster having the highest score will be of rank-1 (high) and so on. Note that we do not consider final tweet representatives as those will keep on changing over time, so a different method is used for cluster ranking. To calculate the scores of tweets, we have used four features: three features are the same as used by *MOOTweetSumm* [10], and the fourth feature uses the concept of named entity recognition (NER) [12]. The used NER identifies the organization, location, numerals, nouns, verbs, and many more named entities. The weighted sum of these features will contribute to the calculation of tweet-score. We showed parameter variation with this four features. At last, two are used. Considering rank-wise clusters from high to low, high scoring tweets are extracted until the desired length of the summary is reached. Note that the existing works [3], [8], [9] suffer from the drawbacks of using different features for summarization. For our experimentation, we have used four different datasets related to disaster events namely, Sandyhook Shooting, UkFlood, and Hagupit, Hyderabad Blast, each having a stream of continuous tweets in a real-time.

Our developed model, *DCBRTS*, is compared with state-of-the-art techniques. Moreover, we have also developed several baselines to reveal the importance of selecting various approaches at different stages of the proposed approach like (a) utilization of three phases in the developed summarization system; (b) to check the suitability of different clustering algorithms in grouping the incoming tweets; (c) to check the suitability of features or existing summarization algorithm to be used for generating the summary in the last phase and so on. For example, in the last phase of real-time tweet summarization, we have explored various existing algorithms like LUHN [13], LexRank [14], TextRank [15], among others. At each phase of the real-time tweet summarization, the used algorithms are shown in the Figure 1. Thus, in total 30 baselines have been developed.

The major contributions of the current paper are listed below: (1) We propose a classification cum real-time tweet summarization, *DCBRTS* system to generate summaries at different breakpoints provided by the user; (2) Designing of deep-learning based classifier to separate situational-tweets from non-situational tweets using their semantics; (3) We model the summarization framework as three phase process: selection of optimal set of tweets using multi-objective optimization concept, clustering, and finally, summary

generation; (4) We have developed the various baselines to compare our systems and to perform in-depth analysis; (5) Extensive experimentations on real-time Twitter dataset illustrate promising results.

The proposed framework is tested on four disaster datasets. The results obtained clearly illustrate the superior performance of our real-time tweet summarization framework over state-of-the-art techniques.

II. RELATED WORKS

During any disaster event, situational tweets serve as useful source for the management authorities. However, these types of tweets need to be extracted properly for their practical utility and to be summarized in real-time. Here, we have discussed the recent techniques developed for classification and summarization.

A. TWEET'S CLASSIFICATION IN DISASTER EVENT

In the literature [16]–[18], several attempts have been made to separate situational tweets from the non-situational. These approaches use the bag-of-words model for classification. However, their performance is heavily dependent on the vocabulary of the disaster events, or in other words, they use in-domain features extracted from the tweets. To overcome the limitation, in [3], authors have developed a classification model that uses domain-independent lexical features to distinguish among tweets. Nowadays, researchers are moving towards building up a deep learning-based classification model. Authors of Caragea *et al.* [5] used a convolution neural network (CNN) [19] for classification of disaster-related tweets. Still, it suffers from the drawback of representing tweets using the bag-of-words model. Recently, the paper by Alrashdi *et al.* [20] developed a bidirectional long short-term memory (Bi-LSTM) [21] model which uses Glove³ word embedding to represent the tweet.

B. REAL-TIME TWEET SUMMARIZATION

Existing research [4], [10], [22]–[26] considered summarization of the available tweets or in other words, those focused only on developing static summarization techniques. However, what is important during a disaster event is real-time summarization of evolving tweet streams. Some of the recent approaches are proposed in [1], [3], [7], [27]. In [1], [27], firstly clustering of tweets is performed and then, representative tweets are selected from each cluster. Finally, ranking of tweets is performed using LexRank [14] algorithm which is a graph-based approach. Rudra *et al.* [3] developed a classification (discussed in previous section) and summarization model. For summarization of situational tweets, they have used the integer linear programming to maximize the number of content words in the summary. Osborne *et al.* [28] proposed a real time event tracking system using greedy summarization. In [7], authors proposed an abstractive summarization method using graph-based scheme. In [29], authors proposed a real-time tweet summarization method

which considers three criteria namely, novelty, informativeness, and relevance with regards to the user's interest for summary generation.

C. ADVANTAGE OVER PRIOR STUDIES

The current work has the following advantages over existing works: 1) To develop a situational tweet classification model, unlike [3] where only syntactic features are used, in the current work, the universal sentence (tweet) vector representation released by Google was used. 2) For summarizing evolving tweet streams, our approach (a) uses the multi-objective evolutionary algorithm to select the optimal set of tweets from the initial collection of tweets; (b) while performing clustering, both syntactic and semantic distances are considered which was absent in [27]; (c) while storing tweets from starting to the end, there may be the problem of storage overhead. This is removed by putting a threshold on the maximum number of tweets, a cluster can retain. This was also done to keep the updated information in the clusters as the tweets arrive; (d) selection of tweets from the final clusters using weighted sum of various newly developed features (different from [3]).

III. CLASSIFICATION MODEL FOR TWEETS

This section focuses on classifying situational tweets from the non-situational tweets using the supervised classifier. For training such classifiers, gold-standard data is required related to the disaster domain. We have collected the annotated data from two sources described below where different human-made and natural disaster events from all over the world are mentioned. Both sources include a variety of disaster events.

A. DATASETS USED FOR CLASSIFICATION

Tweets in the datasets used for classification are divided into several categories and then, they are labeled as situational/non-situational based on their category. The detailed description of annotated data collected from different sources is provided below.

- 1) CrisisLexT26⁴: It includes 1019 unique tweets belonging to different natural disasters like floods, earthquakes, typhoons, haze, and human-made disasters like a terrorist attack, train crashes, explosion, fires, and more. The category-wise statistics for this dataset are shown in Table 2.
- 2) CrisisNLP⁵: It includes a set of 15152 unique tweets related to natural disaster like floods, earthquakes, typhoons, hurricanes, and cyclones and is divided into various categories as shown in Table 3. The label assignment of situational vs. non-situational is also shown.

The distributions of situational vs. non-situational tweets are shown in Table 1. There are some issues in these datasets

³<https://nlp.stanford.edu/projects/glove/>

⁴<https://github.com/sajao/CrisisLex/tree/master/data/CrisisLexT26>

⁵<https://crisisnlp.qcri.org/>

TABLE 1. Data-set statistics for the collected annotated data.

Dataset -> Label->	CrisisLexT26		CrisisNLP	
	Situational	Non-Situational	Situational	Non-Situational
Before Majority voting	6726	3471	13265	1887
After Majority Voting	4037	876	6130	1181
Train Set	2824	613	4291	827
Test Set	1211	263	1839	354

TABLE 2. Different categories under CrisisLexT26 and label assignment.

Category	#Tweets	Label
Caution and advice	1424	Situational
Affected individuals	3072	Situational
Infrastructure and utilities	1114	Situational
Donations and volunteering	1737	Situational
Sympathy and support	2850	Non-Situational

discussed in subsequent sections and resolved using the majority voting concept. The statistics are shown before and after the application of majority voting, a type of ensembling method in machine learning.

B. ISSUES WITH EXISTING DATASETS AND RESOLUTION

Although the above-discussed datasets' annotators have annotated the tweets with only one category, a tweet can sometimes be a mixture of both situational and non-situational segments. For instance, the tweet: *RT live2Tripoli: 400 people have died in the Balochistan earthquake. May God have mercy on all their souls. #Pakistan #Calamity*, is labeled under the injured or missing people category in the CrisisNLP dataset, and hence, it can be considered situational as per the label given. However, the first line (showing causality) of the tweet is situational, while the second line (showing sympathy) is non-situational. We have exploited the majority voting concept to counter this problem and utilized the existing pre-trained SVM classifier [3] trained on (four) disaster events. In this classifier, the tweet is fragmented and then classified into two categories using the lexical features and syntactic features like count of question marks, personal pronouns, exclamations, numerals, intensifiers, and wh-words, etc. (refer to Table-3 of [3]). The training data used in this SVM classifier consisted of very less number of tweets. We aim to develop a generic and efficient classifier using deep learning, which requires a large amount of training data. The

TABLE 3. Different categories under CrisisNLP and label assignment.

Category	#Tweets	Label
other_useful_information	5165	Situational
donation_needs_or_offers	2452	Situational
injured_or_dead_people	2321	Situational
sympathy_and_emotional_support	1887	Non-Situational
caution_and_advice	1011	Situational
infrastructure_and_utilities_damage	1394	Situational
displaced_people_and_evacuations	547	Situational
missing_trapped_or_found_people	375	Situational

descriptions of the steps used for majority voting are provided below:

- 1) We have pre-processed and fragmented the tweets of the collected datasets. After fragmentation, both segments were labeled with the class label as that of the original tweet.
- 2) Utilizing the existing SVM classifier [3], fragmented tweets are classified into either situational or non-situational.
- 3) Finally, if original labels and labels generated using existing SVM classifier are the same, then the tweet is included in the training dataset.

The number of situational and non-situational tweets in each dataset obtained after majority voting is shown in Table 1. The number of tweets in 7:3 ratio for training and testing data, respectively, are also shown in the same table.

C. CLASSIFICATION FEATURES AND MODEL

To make our classifier more robust and efficient than existing classifier [3], we have developed deep learning-based classifier, i.e., *convolution neural network* [19] (with *Sigmoid Focal Cross Entropy* as the loss function (to avoid class imbalance problem)), which was trained on datasets (shown in Table 1) containing tweets from 25 disaster events. Non-situational tweets mostly comprise sentiments like grief, sorrow, hatred, and anger. To capture these sentiments, we have considered semantic features using Google's pre-trained universal sentence encoder model [6].

D. CLASSIFICATION PERFORMANCE

In Table 4, results attained by the developed CNN-based classifier is shown in terms accuracy, precision, recall and F-measure (F-score). Here, we have utilized semantic features (tweet representation using Universal Sentence Encoder). Our proposed model can perform equally well in cross-domain classification. It can be observed that on the

TABLE 4. Classification accuracies, precision, recall, and F1-scores obtained using our developed CNN-based classifier.

Train Dataset ->	CrisisNLP LexT26			
Test Dataset	Accuracy	Precision	Recall	F1-Score
CrisisNLP LexT26	93.06%	94.95%	93.07%	94.00%
CrisisNLP	94.39%	98.93%	94.40%	96.61%
Train Dataset ->	CrisisNLP			
Test Dataset	Accuracy	Precision	Recall	F1-Score
CrisisNLP LexT26	89.63%	86.89%	97.11%	91.72%
CrisisNLP	97.30%	98.19%	98.63%	98.41%

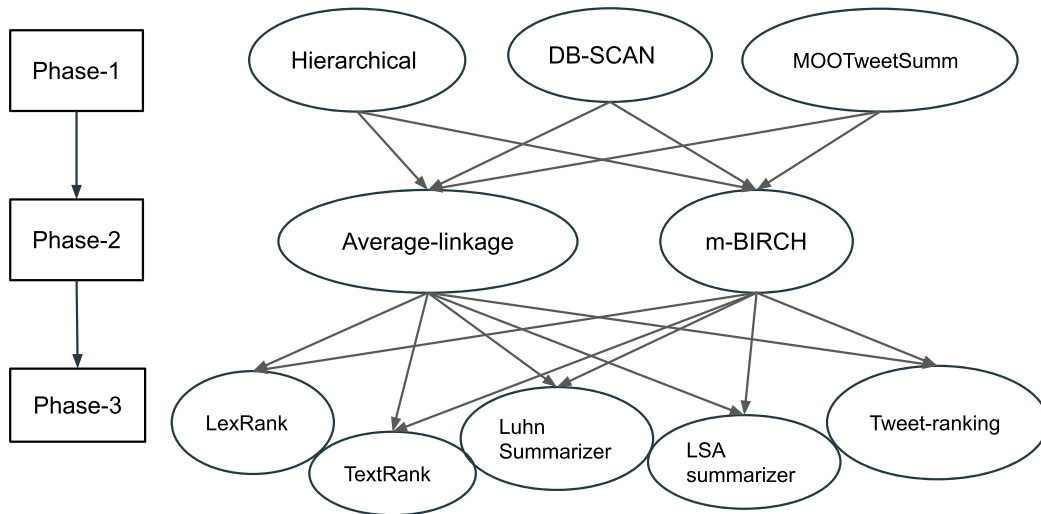


FIGURE 1. Various possibilities at each phase of our framework.

test datasets corresponding to CrisisLexT26 and CrisisNLP, the F-measure value on average is 87% which proves its efficacy for the in-domain datasets. More discussions about our classifier’s performance on the cross-domain datasets are presented in Section VI-A.

IV. REAL-TIME TWEET SUMMARIZATION

Let us assume that we have obtained the set of situational tweets between timestamps t_1 and t_2 ($t_2 \gg t_1$) after applying the classification module. We have solved the process of summarizing tweets in real-time as a three-phase process. First-phase involves initializing the tweet cluster centers using the initial set of tweets. Second-phase occurs when new tweets arrive where they are assigned to the existing clusters or a new cluster is formed. The third phase starts whenever the user requires a summary. This phase comprises of selecting optimal tweets for the final summary. The challenges associated with each phase are discussed in detail in the subsections IV-A, IV-B and IV-C. However, the primary objective is to summarize the tweets in real-time with memory optimization. We have explored several possibilities for each phase, represented in the Figure 1, and described in the relevant sections.

A. INITIALIZATION OF TWEET CLUSTERS

In the first phase, the task is to create tweet clusters using an initial stream of situational tweets (let it be $t_2 - K$ tweets). The performance of our summarization model predominantly depends on this selection process as the clusters will be input to our next phase to determine the final cluster structure. For this purpose, we have used an existing multi-objective optimization based algorithm, namely, MOOTweetSumm [2], discussed below.

1) MOOTweetSumm

This algorithm was designed in the sense that while selecting tweets or a summary, a single tweet may be optimal

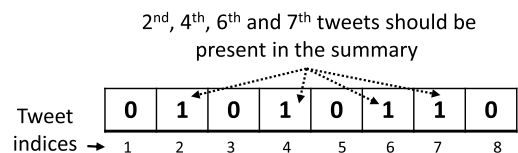


FIGURE 2. Representation of a solution.

considering one perspective but may not be from other perspectives. Therefore, to make the decision process faster, multiple objectives should be considered and those should be optimized simultaneously to select a good set of optimal tweets. Here, we have used two objective functions: tf-idf score of the tweet (Ob1) and anti-redundancy (Ob2) such that

$$\max(Ob1, Ob2) \tag{1}$$

For optimization, a multi-objective binary differential evolution algorithm (MOBDE) [30] is utilized, which is an evolutionary algorithm [31]. It starts from a set of binary solutions where each solution has a maximum length equals to a given set of situational tweets, and the maximum number of ones cannot exceed the desired number of optimal tweets. An example of solution representation is shown in Figure 2 where 1 indicates that the tweet at that index should be in the summary. Each solution is associated with the above two objectives, and MOBDE optimizes these solutions using the iterative procedure. The efficacy of this concept over others is illustrated in the paper [2]. Motivated by this, we have acquired this concept and used it for the selection of the optimal set of tweets.

The selected set of optimal tweets representing the initial stream of tweets will then be utilized as the initial cluster centers. The remaining tweets are assigned to these clusters based on minimum average weighted distance (cosine distance obtained after tf-idf vectorization and Universal Sentence Encoder representation) between a tweet and the cluster center (refer to Eq. 3).

B. UPDATING CLUSTERS

In the second phase, we update the initial clusters formed whenever a new tweet arrives. The new tweet can be merged into an existing cluster whose center is closest to the tweet, or a new cluster will be formed. The main challenge in this phase is finding if the new tweet is similar enough to be merged with the existing cluster. We have used various heuristic approaches where we define a dynamic threshold. If the distance between the cluster center and the new tweet is greater than the threshold, a new cluster is created.

1) m-BIRCH

We have utilized existing *m*-BIRCH (modified-Balanced Iterative Reducing and Clustering Using Hierarchies) clustering algorithm which is recently developed by Madan *et al.* [32]. Noted that *m*-BIRCH is an online clustering algorithm to cluster large datasets in an incremental way and designed to enable data-driven parameter selection and effectively handle differing density reasons. Here, the initial number of clusters equals to the number of optimal tweets selected using MOOTweetSumm, but this can be increased or decreased as the stream of tweets arrive. Let *k*th cluster have $\{t_1, t_2, \dots, t_M\}$ tweets then $Clus_Size_k$ is the size of the *k*th cluster which is calculated as,

$$Clus_Size_k = \begin{cases} 0 & \text{if } M < 1 \\ \sqrt{\frac{2s}{M-1} - \frac{2L^T L}{M(M-1)}} & \text{if } M > 1 \end{cases} \quad (2)$$

where, *M* is the number of tweets in *k*th cluster, *L* is the summation of all tweets vectors, $s = \sum_{i=1}^M \|t_i^v\|_2^2$ is the summation of squares of all the components of the tweet vectors, t_k^v is the vector representation of *k*th tweet. To capture the syntactic and semantic information present in the tweets, we have used the well-known *tf-idf* [33] and recently developed *universal sentence encoder* [6] vector representation, respectively. Therefore, each cluster has two cluster sizes and two cluster centers using syntactic and semantic representations. For example, for *k*th cluster, cluster size is denoted as $Clus_Size_k^1$ (using syntactic) and $Clus_Size_k^2$ (using semantic), while cluster centers are denoted as c_k^1 and c_k^2 . The concept of using two vector representations is like a multi-view learning which states that when the same object (tweet) is seen from any angle then it should belong to the same cluster [34]. When a tweet t_j is to be assigned to *k*th cluster then we consider the average distance as,

$$D(t_j, c_k) = (d_{t_j, c_k^1}^1 + d_{t_j, c_k^2}^2)/2 \quad (3)$$

where, $d_{t_j, c_k^1}^1$ is the cosine distance (1-cosine_similarity) between tweet, t_j , and *k*th cluster center, c_k^1 , in syntactic space. Similarly, $d_{t_j, c_k^2}^2$ is computed in semantic space. When a new tweet, t_m , arrives, its probability of belongingness to the situational category using the developed classification model is first computed. If it belongs to this category, then the following steps are executed

- Find the closest cluster using the shortest average distance criterion (Eq. 3). Let it be the *i*th cluster.
- If $d_{t_j, c_k^1}^1 > (Clus_Size_i^1 \times B)$ or $d_{t_j, c_k^2}^2 > (Clus_Size_i^2 \times B)$, then, a new cluster is created, else, it is merged to the same cluster. Here, *B* is the bounding parameter to control the merging of incoming tweets into the existing clusters. For example, if a cluster is imagined as a sphere with radius *r*, then we want the new tweet to be present in the radius of $r \times B$.
- If the number of tweets in *i*th cluster is greater than *threshold*, then unique *threshold* number of tweets which are closest to the centre are considered. This threshold was kept to reduce information overload and to store updated information.
- If the number of clusters is greater than a *threshold*, two clusters are merged until the number of clusters becomes less than the *threshold*. To determine which clusters should be merged, we determine the distance between two clusters as the sum of the *weighted distances between cluster centers and number of tweets in both clusters divided by the maximum number of possible tweets in a cluster*. This was done to merge those clusters which are semantically similar and have less number of tweets.

C. SUMMARY GENERATION

Whenever a user demands a summary, we have considered the tweets in obtained clusters after the second phase for real-time summary generation instead of considering any window size. Hence, it is possible that there could be some old tweets if they are very informative or useful. In order to do this, we have to select a set of tweets of varying characteristics that contain most of the information. In other words, firstly, the clusters and tweets in each cluster are ranked and then, the top ranked tweets from each cluster are selected in an extractive way considering rank-wise clusters.

1) TWEET-RANKING

Our developed model is based on extractive summarization [10]; hence, from each cluster, tweets are extracted. Therefore ranking of clusters and ranking of tweets in a cluster are required to be performed. For this purpose, firstly, we have computed the tweet's score in each cluster using a weighted sum of four features. Then, the average scores of the tweets belonging to a cluster will be the score of that cluster. Higher the score, the higher will be the rank (rank-1 is considered as the highest). Let *k*th cluster have *M* tweets, $\{t_1, t_2, \dots, t_M\}$, then, tweet-scoring feature for a tweet t_i is described below

- 1) Anti-redundancy ($F1_{t_i}^k$): It is used to remove the redundancy in a summary. For a tweet in the cluster [2], it should be diverse from others in the same cluster; therefore it is computed as

$$F1_{t_i}^k = 2 \times \frac{\sum_{i=1}^M \sum_{j=1, i \neq j}^M D(t_i, t_j)}{M(M-1)} \quad (4)$$

Here, D is the average distance between two tweets in syntactic and semantic space, as described in Eq. (3), $\frac{M(M-1)}{2}$ is the total number of tweet pairs in the same cluster.

- 2) MaxSumTFIDF ($F2_{t_i}^k$): A tweet's score highly depends on relevance of its words [2]. Therefore, we have computed the sum of the tf-idf scores of different words in the tweet, which will be used as the tweet score.
- 3) MaxLength ($F3_{t_i}^k$): In the literature [2], [4], a tweet having maximum length is shown relevant in summary generation. Therefore, this feature is considered into account.
- 4) CountNamedEntities ($F4_{t_i}^k$): In disaster event, named entity recognition plays a major role [9] as it identifies location, organization, numerals, and many more. Therefore, we have counted the number of NERs present in the tweet and divided it by the total number of NERs present in the tweet data to normalize it. Mathematically, it is represented as

$$F4_{t_i}^k = \text{Count}(\text{NER}_{t_i})/Q \quad (5)$$

where, Q is the number of NERs present in the tweet data.

Thus, the final score of tweet t_i in k th cluster will be

$$F_{t_i}^k = \alpha \times \frac{1}{F1_{t_i}^k} + \beta \times F2_{t_i}^k + \gamma \times F3_{t_i}^k + \lambda \times F4_{t_i}^k \quad (6)$$

where, α , β , γ , and λ are the weight factors assigned to different features. Note that for $F2$, $F3$, and $F4$ features, high scores are desired, while for (1), low score is desired. Therefore, to make the weighted sum higher, the value of the feature $F1$ is reversed.

After evaluating tweet's score, high scoring tweets are extracted considering rank-wise clusters, until we get the desired number of tweets in the summary.

V. EXPERIMENTAL SETUP

In this section, we have discussed the datasets, experimental settings, evaluation measures followed by comparative methods.

A. DATASETS

For the purpose of experimentation, we have used four disaster events, including natural and human-made disasters that occurred in different regions of the world. Each dataset is available as a set of 5000 continuous tweet streams with other information like time, date. These datasets are briefly described below:

- 1) Sandyhook Shooting (SHShoot): An assailant killed six adults and 20 children at the Sandy Hook elementary school in Connecticut, USA.
- 2) UkFlood: Landslides and floods in the Uttarakhand state of India.
- 3) Hagupit: A strong cyclone, namely, Typhoon Hagupit, hit the Philippines.

- 4) HyderabadBlast (HBlast): Two bomb blasts in Hyderabad city of India.

The same datasets⁶ are used by the paper [3]. As these datasets are designed for real-time tweet summarization; therefore, gold summaries are provided at two breakpoints of 2000 and 5000 tweets.

B. EXPERIMENTAL SETTINGS

(a) To develop classification model, we have used the sequential model from tensorflow keras framework⁷ with Adam optimizer. Number of epochs used was 20 with early stopping; (b) For selection of representative tweets in section IV-A using MOO-based procedure, we have used the implementation available at the Github repository⁸ with default parameters; (c) The value of bounding factor (B) helping in cluster formulation is kept as 0.6, and the maximum number of tweets (*threshold*) in a cluster should not exceed 40; (d) To identify the NERs present in the tweets, we have used the *spacy*⁹ package of python designed for different natural language processing tasks. Initially, we assume that we have a set of 600 tweets and then, tweets keep on arriving one-by-one.

For rest of the parameters like maximum number representative selection using MOO-based approach, maximum number of clusters and tweets in the clusters, weight factors assigned to different features used in calculating the tweet-score, bounding factor used in clustering, the best values are selected after performing an ablation study as reported in Section VI-C.

C. COMPARATIVE METHODS

For comparison purpose, we have considered COWTS [3] approach for summarizing disaster specific events in real-time. COWTS focused on extracting tweets having the maximum number of content words (nouns, numerals, and verbs). Note that this approach also classifies tweets as situational or non-situational and then summarizes situation tweets. But, in comparison to ours, it is not efficient in terms of memory optimization as it does not discard any situational tweets. In addition to COWTS, we have developed several baselines of our proposed approach, *DCBRTS*, by varying methodologies used in different stages/phases, to prove its efficacy. The possible baselines are graphically shown in Figure 1 and discussed below:

- Initialization of tweet clusters (Phase 1): As discussed in section IV-A, the initial stream of tweets are clustered. We have used two well-known clustering algorithms for the same.
 - DBSCAN¹⁰: Density-based clustering is most commonly used non-parametric algorithm. Given a set

⁶http://cse.iitkgp.ac.in/krudra/disaster_dataset.html

⁷<https://www.tensorflow.org/guide/keras>

⁸https://github.com/nsaini1988/Microblog_Summarization

⁹<https://spacy.io/usage/linguistic-features>

¹⁰<https://en.wikipedia.org/wiki/DBSCAN>

of points in some space, it groups points closely packed together, identifying points lying alone in low-density regions as outlier points. DBSCAN requires a parameter eps which was set to 0.2 and $min\ samples$ equal to 5.

- Hierarchical Clustering¹¹: It is a cluster analysis method that explores to build a hierarchy of clusters. We have used the agglomerative bottom-up approach, where each observation is initially its cluster. Then, moving up in the hierarchy, the pairs of clusters are merged. It requires number of clusters to be created as a parameter which was set to number of clusters initially created by DBSCAN.
- Updating Clusters (Phase-2): We have devised another approach to determine if a new tweet should be merged in the existing cluster or a new cluster is to be formed. As compared to the m-BIRCH, the difference lies in the determination of cluster size. Here, we have defined it as the average of the cosine distance from the closest cluster center to other cluster's center. Let the cluster centers be $\{t_1, t_2, \dots, t_M\}$ and M be total number of clusters, then $Clus_Size_k$ is the size of the k th cluster which is calculated as,

$$Clus_Size_k = \left(\sum_{i=1}^M d_{i,k} \right) / M \quad (7)$$

where, $d_{i,k}$ is the cosine distance (1-cosine_similarity) between tweet center, t_i , and t_k cluster center, c_k^1 , in syntactic space. Similarly, $d_{t_j, c_k^2}^2$ is computed in semantic space.

- Summary Generation (Phase-3): We employed existent summarization approaches where all the tweets in the clusters are passed as input to the approach. The approaches used are presented below:
 - LexRank¹²: LexRank is an unsupervised graph based commonly used approach for automatic text summarization where graph method is exploited to score sentences.
 - TextRank¹³: It uses similarity of one sentence to all other sentences. A sentence, which is the most similar to all the other sentences, is considered the most important sentence.
 - Luhn Summarizer¹⁴: It is a naive summarization approach based on TF-IDF and sentences are ranked based on keyword frequency and proximity within a sentence.
 - LSA Summarizer¹⁵: Latent Semantic Analysis is a relatively new algorithm which combines term frequency with singular value decomposition.

¹¹https://en.wikipedia.org/wiki/Hierarchical_clustering

¹²<https://pypi.org/project/lexrank/>

¹³<https://pypi.org/project/textrank/>

¹⁴https://en.wikipedia.org/wiki/Sentence_extraction

¹⁵<http://www.kiv.zcu.cz/jstein/publikace/isim2004.pdf>

D. EVALUATION MEASURE

For evaluating the quality of the summary, we have used the well-known ROUGE-N metric, which counts the N-gram overlapping words between predicted and actual summary. More specifically, we actually used ROUGE-L F-score which is longest Common Subsequence (LCS) based statistics. Note that baseline papers reported ROUGE-1 F-score as an evaluation measure. However, by just using Rouge-1, we are only scoring whether single words overlap in the predicted output and the ground truth. As all tweets are around a particular topic, this seems to be a straightforward objective. Hence we have utilized ROUGE-L F-score.

VI. EXPERIMENTAL RESULTS

This Section will describe the results of our two-phase dynamic summarization approach on the datasets discussed in Section V-A. Note that the authors of *COWTS* method have also developed a classification cum summarization technique; therefore, we have executed the code of *COWTS* to obtain the results.

A. CLASSIFICATION RESULTS

To evaluate the developed CNN-based classification model's performance, we have selected the cross-domain datasets discussed in Section V-A. Note that these datasets are not used as parts of the training data. As can be analyzed from Table 5, our classifier's performance is also better on the cross-domain datasets because the training dataset consists of various natural and man-made disaster-related information. On the other hand, using the existing SVM classifier [3] utilizing lexical and syntactic features, average accuracy over four datasets was reported as 79.5% (reported in [3]), which is relatively less in comparison to our classifier. This proves the efficacy of our deep learning-based classifier over existing classifier.

TABLE 5. Cross-domain classification results on CNN models.

Train Dataset ->	CrisisLexT26		CrisisNLP	
Test Dataset	Accuracy	F1-Score	Accuracy	F1-Score
Hagupit	81.54%	86.06%	79.21%	85.06%
HBlast	86.61%	89.52%	74.28%	81.80%
SHShoot	91.16%	92.77%	85.12%	88.50%
UkFlood	78.74%	81.51%	70.36%	77.40%

B. REAL-TIME SUMMARIZATION RESULTS

In Table 6, a comparison of our proposed algorithm for real-time tweet summarization is shown with *COWTS*, at two breakpoints of 2000 and 5000 tweets in terms of Rouge-L F-score. Since the baseline paper *COWTS* reported ROUGE-1 F-Score, we executed the code of *COWTS* again to get the Rouge-L F-score. All algorithms are unsupervised in nature. It is evident that our approach performs better than existing ones. For instance, considering mean Rouge-L F-score over all datasets, our method improves by 4% over *COWTS*. The higher scores attained by our approach over *COWTS* indicate that our

TABLE 6. Comparison of Rouge-L F-score obtained by our real-time summarization approach and existing approach using situational tweet streams at breakpoints of 2000 and 5000 tweets.

Breakpoints Datasets ->	ROUGE-L F-score							
	HBlast		UkFlood		SHShoot		Hagupit	
	Proposed	COWTS	Proposed	COWTS	Proposed	COWTS	Proposed	COWTS
0-2000	0.5674	0.5610	0.3702	0.3412	0.5051	0.4645	0.3446	0.3013
2000-5000	0.3947	0.3824	0.2398	0.2341	0.4566	0.2911	0.3556	0.3372
Average Scores	0.4811	0.4717	0.3050	0.2877	0.4808	0.3778	0.3501	0.3193

TABLE 7. ROUGE-L F-score obtained by DBSCAN and m-Birch combination using situational tweet streams at breakpoints of 2000 and 5000 tweets.

Dataset	Breakpoint	Summarization Algorithm				
		TextRank	LexRank	LSA Summarizer	Luhn Summarizer	Tweet-Ranking
SHShoot	2000	0.4224	0.4715	0.4639	0.5087	0.5068
	5000	0.4087	0.4248	0.4277	0.4388	0.4646
HBlast	2000	0.2394	0.3142	0.4257	0.4157	0.4830
	5000	0.2984	0.2081	0.2960	0.2525	0.3611
UKFlood	2000	0.3881	0.2300	0.2793	0.2667	0.2814
	5000	0.2358	0.1950	0.1411	0.2407	0.2654
Hagupit	2000	0.2014	0.1766	0.2818	0.2404	0.2494
	5000	0.1805	0.1677	0.2500	0.1667	0.2252

TABLE 8. ROUGE-L F-score obtained by MooTweetSumm and m-Birch combination using situational tweet streams at breakpoints of 2000 and 5000 tweets.

Dataset	Breakpoint	Summarization Algorithm				
		TextRank	LexRank	LSA Summarizer	Luhn Summarizer	Tweet-Ranking
SHShoot	2000	0.4444	0.4245	0.3938	0.4966	0.5051
	5000	0.3735	0.3777	0.3951	0.4281	0.4566
HBlast	2000	0.2857	0.3321	0.4869	0.3232	0.5674
	5000	0.2810	0.2626	0.2419	0.2486	0.3947
UKFlood	2000	0.3493	0.2168	0.2805	0.2120	0.3702
	5000	0.2436	0.1893	0.1314	0.2444	0.2398
Hagupit	2000	0.1861	0.1634	0.2159	0.2232	0.3446
	5000	0.1675	0.1420	0.1996	0.1632	0.3556

TABLE 9. ROUGE-L F-score obtained by hierarchical and m-Birch combination using situational tweet streams at breakpoints of 2000 and 5000 tweets.

Dataset	Breakpoint	Summarization Algorithm				
		TextRank	LexRank	LSA Summarizer	Luhn Summarizer	Tweet-Ranking
SHShoot	2000	0.4068	0.5020	0.4386	0.5050	0.4132
	5000	0.3642	0.4184	0.4037	0.3878	0.4195
HBlast	2000	0.2976	0.3185	0.5066	0.3187	0.3810
	5000	0.2754	0.2585	0.2227	0.2426	0.2687
UKFlood	2000	0.3842	0.2330	0.2819	0.2747	0.2487
	5000	0.2480	0.1937	0.1623	0.2627	0.2424
Hagupit	2000	0.2312	0.1830	0.2523	0.2089	0.2708
	5000	0.1845	0.1613	0.2428	0.1822	0.2313

three-phase dynamic summarization system, i.e., selection of representative tweets using multi-objective optimization, m-BIRCH online clustering algorithm, and summary generation using various features, along with CNN-based situational tweet selection approach, are better for generating real-time summary.

We have presented detailed results over individual datasets at both 2000 and 5000 breakpoints obtained using different

variant of our proposed approach, *DCBRTS* in Tables 7 to 12. Note that our proposed model is a three-phase model where we have explored 3, 2, 5 possibilities (changing the working scenario) in the first, second, and third phase model, respectively. Each table shows the ROUGE-L F1-Score obtained for a combination of possibility from the first and second phases. The average results over all datasets are reported in in Table 13.

TABLE 10. ROUGE-L F-score obtained by DBSCAN and average-linkage combination using situational tweet streams at breakpoints of 2000 and 5000 tweets.

Dataset	Breakpoint	Summarization Algorithm				
		TextRank	LexRank	LSA Summarizer	Luhn Summarizer	Tweet-Ranking
SHShoot	2000	0.4187	0.4427	0.4737	0.4818	0.4830
	5000	0.3976	0.4511	0.3988	0.4281	0.3672
HBlast	2000	0.2749	0.3650	0.4863	0.3440	0.4543
	5000	0.2793	0.2562	0.2364	0.2332	0.2644
UKFlood	2000	0.4089	0.2452	0.2811	0.2421	0.3274
	5000	0.2425	0.1790	0.1725	0.2319	0.2482
Hagupit	2000	0.2211	0.1602	0.2406	0.2594	0.2843
	5000	0.2004	0.1502	0.1721	0.1717	0.2163

TABLE 11. ROUGE-L F-score obtained by MOOTweetSumm and average-linkage combination using situational tweet streams at breakpoints of 2000 and 5000 tweets.

Dataset	Breakpoint	Summarization Algorithm				
		TextRank	LexRank	LSA Summarizer	Luhn Summarizer	Tweet-Ranking
SHShoot	2000	0.4290	0.4444	0.4444	0.4865	0.4684
	5000	0.3965	0.4603	0.3877	0.3782	0.4088
HBlast	2000	0.3020	0.3320	0.4894	0.3242	0.4405
	5000	0.3211	0.2483	0.2342	0.2329	0.3627
UKFlood	2000	0.3570	0.2148	0.2844	0.2482	0.3618
	5000	0.2557	0.1896	0.1598	0.2391	0.1910
Hagupit	2000	0.1822	0.1534	0.1871	0.2248	0.2244
	5000	0.1643	0.1388	0.1832	0.1535	0.2084

TABLE 12. ROUGE-L F-score obtained by hierarchical and average-linkage combination using situational tweet streams at breakpoints of 2000 and 5000 tweets.

Dataset	Breakpoint	Summarization Algorithm				
		TextRank	LexRank	LSA Summarizer	Luhn Summarizer	Tweet-Ranking
SHShoot	2000	0.4226	0.4506	0.4248	0.4679	0.5032
	5000	0.4154	0.4444	0.3988	0.3861	0.4281
HBlast	2000	0.2808	0.3320	0.4974	0.3171	0.4335
	5000	0.2568	0.2617	0.2466	0.2481	0.1940
UKFlood	2000	0.3981	0.2413	0.2731	0.2782	0.3832
	5000	0.2375	0.1802	0.1639	0.2371	0.2100
Hagupit	2000	0.2090	0.1590	0.2333	0.2446	0.2737
	5000	0.2004	0.1468	0.2134	0.1653	0.2036

TABLE 13. Comparison of Rouge-L F-score obtained by various baselines. Rows are labelled using clustering algorithm used in phase 1 and phase2. Columns represent summarization algorithms used in phase-3.

Phase1_Phase2	Phase3				
	TextRank	LexRank	LSA Summarizer	Luhn Summarizer	Tweet-Ranking
DBSCAN_m-Birch	0.2968	0.2735	0.3207	0.3163	0.3546
MOOTweetSumm_m-BIRCH	0.2914	0.2635	0.2931	0.2924	0.4042
Hierarchical_m-Birch	0.2990	0.2836	0.3139	0.2978	0.3094
DBSCAN_Average-linkage	0.3054	0.2812	0.3077	0.2990	0.3306
MOOTweetSumm_Average-linkage	0.3010	0.2727	0.2963	0.2859	0.3333
Hierarchical_Average-linkage	0.3026	0.2770	0.3064	0.2930	0.3287

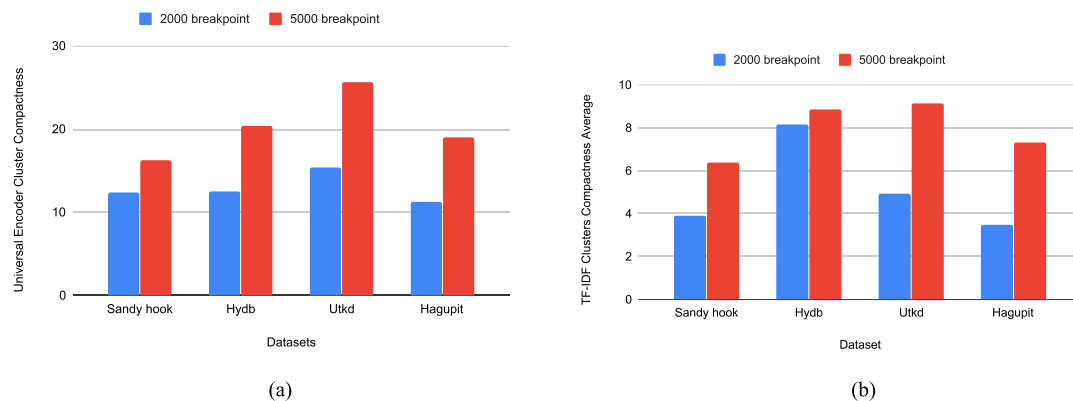
To illustrate the nature of summary, we have shown an example of generated summary in Table 14 for *HBlast* dataset at a breakpoint of 2000 tweets, in comparison with corresponding gold summary. The matched lines are shown by same colours (excluding black colour). We have only highlighted complete tweets which are occurred in both the summaries. This generated system has Rouge-L F-score score of 0.56.

To check the cluster qualities at different breakpoints, we have plotted the average of the compactness of the clusters at both the breakpoints. Note that we have utilized both universal encoder (semantic) and tf-idf (syntactic) representation while calculating distance (refer to Eq. 3). As each cluster is expected to have maximum of *threshold* number of tweets to avoid information overload due to continuous tweet streaming and cosine distance can have value between 0 to 2,

TABLE 14. This table contains one of the best system summaries generated by our proposed approach for Hblast dataset and gold summary at 5000 breakpoint.

Predicted Summary: Death toll now 12 , injured 57 , Home Secretary RK Singh . Police Control Rooms , +91-40-27852435-36 , +91-40-23261166 . Police suspect one of the bombs may have been kept on a motorcycle , the other in a tiffin box . BREAKING , Twin blast in Hyderabad's Dilsukh Nagar suburb , reports of 15 deaths , over 50 injured . Some more bombs recovered at Bus stop and a Foot over bridge , place not mentioned . One blast took place near Venkatadri theatre , second near Konark Theatre . Needs AB+ve blood For , Farida Narayana Hrudayalaya , Suraram , Jeedimetla , Call , 9676595836 . Dilshuknagar Hospitals , Sigma 40-67120218 , Good Life 49640328 , Vasan eye 43400200 , HariPrasad 2404673 . Two deadly blasts in Dilsukh Nagar in Hyderabad , several casualties , 50 reported injured with 10 people died . List of hospitals in Dilsukh Nagar , Hyderabad . UPDATE , AFP , police say seven people have died and 47 people hurt in bomb blasts in Indian city of Hyderabad . 15 killed 50 injured in Hyderabad blast More Photos . 2 blasts reported near bus stand in southern Indian city of Hyderabad , 10 people feared dead , at least 40 others . Emergency helpline number of the Andhra Pradesh government 040-27854771 . Hyderabad blasts , IB , NIA , NSG officials to reach explosion sites soon , says RK Singh . Blood banks Dilsuknagar , SLMS 040-64579998 , Kamineni 39879999 , Hima Bindu 9246373536 , Balaji 2457219 . Lashkar announces unconditional surrender after Manmohan Singh strongly condemns the HyderabadBlasts . Share it Hospitals Hyderabad Blasts , 91)-40-6711514 , 8Ikon Hospital , Paramitha . 11 killed in Hyderabad blast .

Gold Summary: Blood banks Dilsuknagar , SLMS 040-64579998 , Kamineni 39879999 , Hima Bindu 9246373536 , Balaji 2457219 . Dilshuknagar Hospitals , SaiRam 04024064532 , Vijaya 24069500 , Savitha 66632381 . Dilshuknagar Hospitals , Sigma 40-67120218 , Good Life 49640328 , Vasan eye 43400200 , HariPrasad 2404673 . State Helplines for Hyderabad Blasts , 040 27854771 , 040-27853408 040 27852435-36 . Share it Hospitals Hyderabad Blasts , 91)-40-6711514 , 8Ikon Hospital , Paramitha . Help Numbers , Dhanalaxmi Ambulance Services at Dilshuknagar , +91 9391351543 , 9963857749 , 9440379926 . One blast took place near Venkatadri theatre , second near Konark Theatre . Hyderabad , Blast Needs A-ve blood , 3 units , For , Vishwanath At , Narayana Hrudayalaya , Suraram , Jeedimetla . Needs AB+ve blood For , Farida Narayana Hrudayalaya , Suraram , Jeedimetla , Call , 9676595836 . Alert in all major cities Mumbai , Kerala , Karnataka , Delhi across India . Some more bombs recovered at Bus stop and a Foot over bridge , place not mentioned . Hyderabad blast , 12 killed in Hyderabad blast An injured person is treated at the Omini hospital Kothapet in . NIA , NSG teams flying to Hyderabad blast site . List of hospitals in Dilsukh Nagar , Hyderabad . 7 allegedly died , 20 injured . 2 blasts reported near bus stand in southern Indian city of Hyderabad , 10 people feared dead , at least 40 others . 9 killed , 32 injured in serial blasts in Hyderabad . Police say 12 dead , 52 injured in two bomb blasts in Hyderabad . Shattered glass , blood , slippers strewn at the blast spot in Dilsukh Nagar in Hyderabad . Twin blast in Hyderabad's Dilsukh Nagar reports of 15 deaths , over 50 injured . 9 killed in Hyderabad blast , 5 in police firing . Lot of traffic moving around and 7 confirmed dead . Indian interior minister says 11 people killed .

**FIGURE 3.** Average of the compactness of the clusters obtained at two breakpoints for all datasets using a) Universal sentence encoder (semantic tweet representation); b) tf-idf (syntactic representation).**TABLE 15.** Effect of B on the number of merge operations (merging two clusters) and addition operations (adding a new tweet in the clusters) performed during clustering.

B	#cnt_merges	#cnt_additions
0.0	2824	49
0.2	1611	1262
0.4	731	2142
0.6	73	2800
0.8	0	2907

maximum compactness for a cluster is expected to be $threshold * 2$. The average compactness of clusters formed at 2000 and 5000 breakpoints for all datasets are shown in Figure 3. In this Figure, part (a) and (b) illustrate the average compactness using semantic and syntactic representations, respectively. The values shown clearly suggest that clusters created are compact and are of good quality.

C. SENSITIVITY ANALYSIS

In this section, we have shown the sensitivity analysis on various parameters like bounding factor, maximum number of clusters, and many more.

1) EFFECT OF BOUNDING FACTOR (B)

We have used B as a bounding factor while updating the clusters. If B is small, then the probability of merging the new tweet into the existing cluster decreases, resulting in the formation of new clusters. If a large number of new clusters are created, it will be computationally expensive to merge them. On the other hand, if B is large, most of the tweets will be absorbed by the existing clusters affecting cluster quality and compactness. Table 15 shows the number of operations done when two clusters are merged into one cluster (cnt_merges) and the number of times when new tweets are added into the

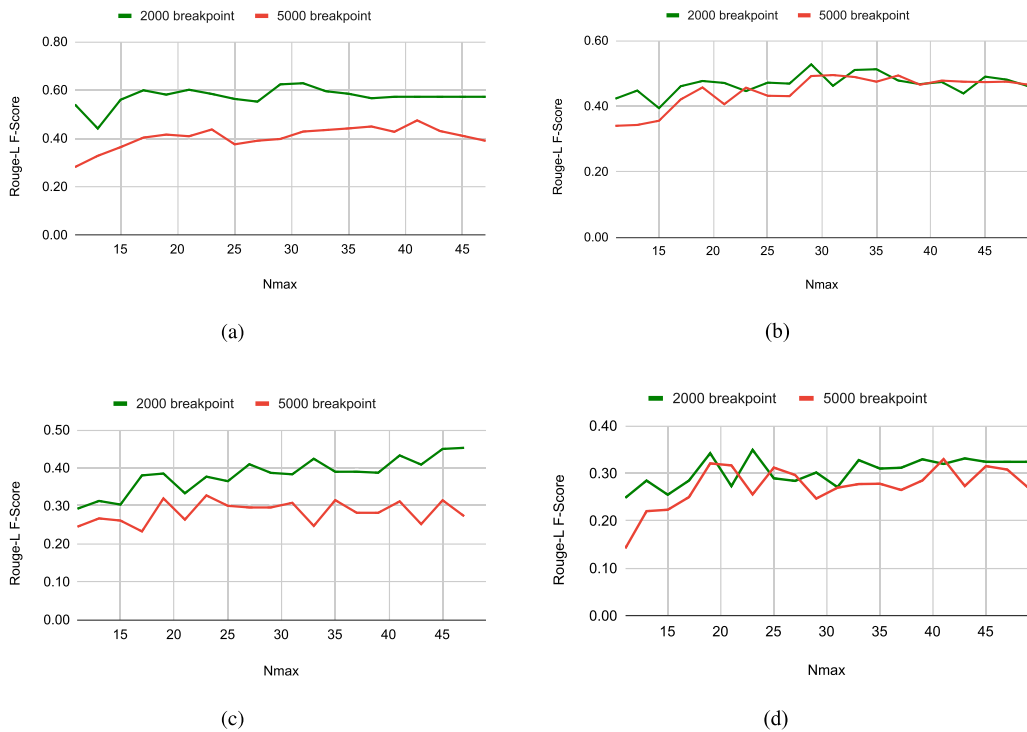


FIGURE 4. Effect of maximum number of clusters on Rouge-L F-score for (a) HBLast, (b) SHShoot, (c) UkFlood, and (d) Hagupit datasets.

existing cluster (*cnt_additions*). Here, $B = 0.6$ is shown to have a good balance.

2) EFFECT OF N_{max}

Figure 4 depicts the change of Rouge-L F-Score with change in maximum number of clusters. When N_{max} is small, Rouge-L F-Score is less due to substantial loss of information. When N_{max} is too large, clustering becomes slow due to large number of clusters. Also, storage overhead is higher for large N_{max} . A balanced value for N_{max} is 40 which is used for generating results.

3) EFFECT OF MAXIMUM NUMBER OF TWEETS IN A CLUSTER

As maximum number of tweets in a cluster increases, more information is stored in a cluster and summary quality improves. Large value of the same results in information dissipation which can be missed by summarization algorithm. Also, it increases computations and storage overhead. As shown in Figure 5, 40 is selected as the optimal value of this parameter.

4) EFFECT OF NUMBER OF TWEETS USED FOR CREATING CLUSTERS

Initial number of tweets used for creating clusters determines the cluster stream quality for further processing. Though the summary quality remains almost similar with change in the number as shown in Figure 6, we selected 600 tweets for initial clustering to ensure good dynamic partitioning is created.

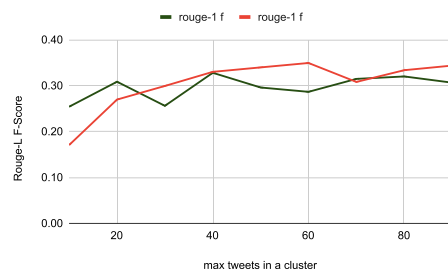


FIGURE 5. Effect of maximum number of tweets in a cluster.

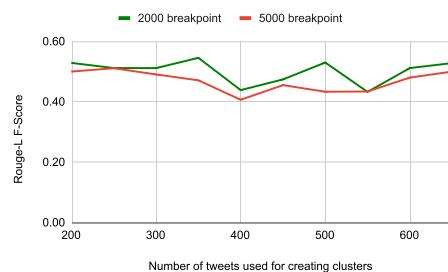


FIGURE 6. Effect of number of tweets used for initial clustering.

5) ABLATION STUDY FOR TWEET-SCORING FEATURES

An ablation study for weight factors (α , β , γ , and λ) assigned to various tweet-scoring features (*Anti - redundancy*, *MaxSumTFIDF*, *MaxLength*, and *CountNamedEntities*) is shown in Table 16. From this Table, it is evident that *MaxSumTFIDF* and *CountNamedEntities* features, both

TABLE 16. Ablation study for weight factors used with tweet-scoring features in summary generation.

Rouge-L F-Score		Weight Factors			
2000 breakpoint	5000 breakpoint	γ	α	β	λ
0.43	0.36	1.00	0.00	0.00	0.00
0.43	0.38	0.00	1.00	0.00	0.00
0.48	0.45	0.00	0.00	1.00	0.00
0.40	0.37	0.00	0.00	0.00	1.00
0.47	0.43	0.25	0.25	0.25	0.25
0.45	0.44	0.00	0.33	0.33	0.33
0.48	0.47	0.33	0.00	0.33	0.33
0.43	0.38	0.33	0.33	0.00	0.33
0.46	0.44	0.33	0.33	0.33	0.00
0.51	0.46	0.00	0.00	0.50	0.50
0.36	0.32	0.00	0.50	0.00	0.50
0.50	0.43	0.00	0.50	0.50	0.00
0.43	0.30	0.50	0.00	0.00	0.50
0.49	0.44	0.50	0.00	0.50	0.00
0.44	0.37	0.50	0.50	0.00	0.00

having equal weightage of 0.5, helped in increasing the summary quality as the number of arriving new tweets increases. Hence, these features with equal weight-ages are considered for summary generation in the reported results.

VII. CONCLUSION

The current article presents a novel framework for classification followed by summarization to handle the continuous tweet streams posted during disaster events. This system can help the disaster management authorities to perform the immediate relief operation. For classification, a deep learning-based classifier is proposed, which identifies the situational tweets using semantic and syntactic features. The concept of ensemble learning (majority voting) is also utilized which takes help of existing classifiers to design such classifier. The identified situational tweets are then used as inputs to the real-time summarization system. We have derived various key-insights from the developed summarization framework: (a) selection of representative tweets from the initial set of situational tweets using multi-objective evolutionary algorithm helps in providing a right direction for optimal summary formulation; (b) the use of online clustering algorithm helps in clustering the incoming tweets and by putting a threshold on the maximum number of clusters and the maximum number of tweets in a cluster help in minimizing information overload; (c) tf-idf score and count of named entities, both together help in generating better summary than COWTS and Sumblr. In terms of improvement, considering mean Rouge-L F-score over all datasets, our method improves by 4% on an average over COWTS.

In the future, we would like to extend the work for sentiment aware real-time microblog classification-summarization framework and its application to multiple regional languages so that it can be beneficial to the community.

REFERENCES

[1] Z. Wang, L. Shou, K. Chen, G. Chen, and S. Mehrotra, "On summarization and timeline generation for evolutionary tweet streams," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1301–1315, May 2015.

[2] N. Saini, S. Saha, and P. Bhattacharyya, "Multiobjective-based approach for microblog summarization," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 6, pp. 1219–1231, Dec. 2019.

[3] K. Rudra, S. Ghosh, N. Ganguly, P. Goyal, and S. Ghosh, "Extracting situational information from microblogs during disaster events: A classification-summarization approach," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 583–592.

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

[5] C. Caragea, A. Silvescu, and A. H. Tapia, "Identifying informative messages in disaster events using convolutional neural networks," in *Proc. Int. Conf. Inf. Syst. Crisis Response Manage.*, 2016, pp. 137–147.

[6] D. Cer, Y. Yang, S.-Y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," 2018, *arXiv:1803.11175*. [Online]. Available: <http://arxiv.org/abs/1803.11175>

[7] A. Olariu, "Efficient online summarization of microblogging streams," in *Proc. 14th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 2, 2014, pp. 236–240.

[8] K. Rudra, P. Goyal, N. Ganguly, S. Ghosh, and M. Imran, "Extracting and summarizing situational information from the Twitter social media during disasters," *ACM Trans. Web*, vol. 12, no. 3, p. 17, 2018.

[9] K. Rudra, P. Goyal, N. Ganguly, P. Mitra, and M. Imran, "Identifying sub-events and summarizing disaster-related information from microblogs," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 265–274.

[10] N. Saini, S. Saha, A. Jangra, and P. Bhattacharyya, "Extractive single document summarization using multi-objective optimization: Exploring self-organized differential evolution, grey wolf optimizer and water cycle algorithm," *Knowl.-Based Syst.*, vol. 164, pp. 45–67, Jan. 2019.

[11] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[12] K. Rudra, A. Sharma, N. Ganguly, and S. Ghosh, "Characterizing and countering communal microblogs during disaster events," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 2, pp. 403–417, Jan. 2018.

[13] H. P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Develop.*, vol. 2, no. 2, pp. 159–165, 1958.

[14] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," *J. Artif. Intell. Res.*, vol. 22, pp. 457–479, Dec. 2004.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep., 1999.

[16] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, "AIDR: Artificial intelligence for disaster response," in *Proc. 23rd Int. Conf. World Wide Web*, Apr. 2014, pp. 159–162.

[17] S. Verma, S. Vieweg, W. J. Corvey, L. Palen, J. H. Martin, M. Palmer, A. Schram, and K. Anderson, "Natural language processing to the rescue? Extracting 'situational awareness' tweets during mass emergency," in *Proc. 5th Int. AAAI Conf. Weblogs Social Media*, 2011, pp. 385–392.

[18] I. Varga, M. Sano, K. Torisawa, C. Hashimoto, K. Ohtake, T. Kawai, J.-H. Oh, and S. De Saeger, "Aid is out there: Looking for help from tweets during a large scale disaster," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2013, pp. 1619–1629.

[19] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: <http://arxiv.org/abs/1408.5882>

[20] R. ALRashdi and S. O'Keefe, "Deep learning and word embeddings for tweet classification for crisis response," 2019, *arXiv:1903.11024*. [Online]. Available: <http://arxiv.org/abs/1903.11024>

[21] N. K. Nguyen, A.-C. Le, and H. T. Pham, "Deep bi-directional long short-term memory neural networks for sentiment analysis of social data," in *Proc. Int. Symp. Integr. Uncertainty Knowl. Modeling Decis. Making*, Springer, 2016, pp. 255–268.

[22] 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.

[23] J. Bian, Y. Yang, and T.-S. Chua, "Multimedia summarization for trending topics in microblogs," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2013, pp. 1807–1812.

- [24] R. Wang, S. Luo, L. Pan, Z. Wu, Y. Yuan, and Q. Chen, "Microblog summarization using paragraph vector and semantic structure," *Comput. Speech Lang.*, vol. 57, pp. 1–19, Sep. 2019.
- [25] N. Avudaiappan, A. Herzog, S. Kadam, Y. Du, J. Thatche, and I. Safro, "Detecting and summarizing emergent events in microblogs and social media streams by dynamic centralities," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 1627–1634.
- [26] K. Mehra and V. Chandra, "Summarizing microblogs for emergency relief and preparedness," in *Proc. SMERP@ECIR, 2017*, pp. 104–108.
- [27] L. Shou, Z. Wang, K. Chen, and G. Chen, "Sumblr: Continuous summarization of evolving tweet streams," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2013, pp. 533–542.
- [28] M. Osborne, S. Moran, R. McCreadie, A. Von Lunen, M. Sykora, E. Cano, N. Ireson, C. Macdonald, I. Ounis, Y. He, T. Jackson, F. Ciravegna, and A. O'Brien, "Real-time detection, tracking, and monitoring of automatically discovered events in social media," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations, 2014*, pp. 37–42.
- [29] A. Chellal, M. Boughanem, and B. Dousset, "Multi-criterion real time tweet summarization based upon adaptive threshold," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Oct. 2016, pp. 264–271.
- [30] 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.
- [31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [32] S. Madan and K. J. Dana, "Modified balanced iterative reducing and clustering using hierarchies (m-BIRCH) for visual clustering," *Pattern Anal. Appl.*, vol. 19, no. 4, pp. 1023–1040, Nov. 2016.
- [33] J. Ramos, "Using TF-IDF to determine word relevance in document queries," in *Proc. 1st Instruct. Conf. Mach. Learn.*, vol. 242, 2003, pp. 133–142.
- [34] S. Mitra, M. Hasanuzzaman, and S. Saha, "A unified multi-view clustering algorithm using multi-objective optimization coupled with generative model," *ACM Trans. Knowl. Discovery Data*, vol. 14, no. 1, pp. 1–31, Feb. 2020.



DIKSHA BANSAL received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Patna. Her research interests include deep learning, natural language processing, and explainable AI.



NAVEEN SAINI (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Indian Institute of Technology, Patna, India. He is currently an Assistant Professor with the Endicott College of International Studies, Woosong University, Republic of Korea. Before this, he was a Postdoctoral Fellow at the Institut De Recherche En Informatique De Toulouse (IRIT), which is a joint research unit of the Université Toulouse

III—Paul Sabatier, Toulouse, France. His current research interests include developing algorithms for text clustering and automatic summarization systems using machine learning, multi-objective optimization, and evolutionary algorithms. His area of research broadly covers many facets of extractive summarization (<https://sites.google.com/view/nsaini>).



SRIPARNA SAHA (Senior Member, IEEE) received the master's and Ph.D. degrees in computer science from the Indian Statistical Institute, Kolkata, India, in 2005 and 2011, respectively. She is currently an Associate Professor with the Department of Computer Science and Engineering, Indian Institute of Technology, Patna, India. Her current research interests include machine learning, pattern recognition, multi-objective optimization, language processing, and biomedical information extraction. She has authored or coauthored more than 120 papers. She was a recipient of various prestigious awards, like Google India Women in Engineering Award, in 2008; the NASI Young Scientist Platinum Jubilee Award, in 2016; the BIRD Award, in 2016; and the IEI Young Engineer's Award, in 2016 (<https://www.iitp.ac.in/sriparna/>).

• • •