GenSumm: A Joint Framework for Multi-Task Tweet Classification and Summarization Using Sentiment Analysis and Generative Modelling

Diksha Bansal, Rahul Grover, Naveen Saini, *Member, IEEE*, and Sriparna Saha, *Senior Member, IEEE*

Abstract—Social media platforms like Twitter act as the medium for communication among people, government agencies, NGOs, and other relief providing agencies in widespread humanitarian havoc during a disaster outbreak when other communication means might not be available. Various agencies leverage Twitter's open and public features to get timely and reliable updates, thus support agencies in communicating with the people on rescue and provide immediate relief. As situational updates are mixed in millions of other tweets, an efficient system is required to extract and summarize these tweets. We have developed a novel framework that uses a deep learning-based classification model to separate the informational tweets from others and summarizes them in the current paper. Non-situational tweets mostly comprise sentiments like grief, anger, sorrow, etc. Motivated by this observation, we have solved sentiment classification and informative tweet selection tasks simultaneously using a multi-task learning (MTL) in a deep-learning framework. Our summarization approach generates clustering solutions using various existing approaches and then ensembles cluster solutions using generative modelling. A summary is formulated by extracting tweets from different clusters. The proposed approach's superior performance on four disaster-related events indicates the developed framework's efficiency over the state-of-the-art techniques.

Index Terms—Summarization, classification, convolution neural network, clustering, generative modelling

1 Introduction

The popularity of social media platforms like Twitter has been increasing and has transformed the world around us[1,2]. With millions of tweets being posted daily, Twitter has become an essential source of real-time information [1]. Users of the platform benefit from it by getting coverage of what's happening around in sectors like health, education, politics, sports, etc. However, sometimes it might become tiresome to extract out meaningful or the required information from such a large number of tweets just by using the search feature. It can be well understood that information extracted from these tweets, especially in a disaster or calamity, can be beneficial for disaster management authorities to conduct relief operations with maximum impact [2], [3], [4].

At the time of a calamity, a plethora of information is posted rapidly on such social networking platforms. It can be well imagined that filtering out useful information during such difficult times is essential and, at the same time, an arduous task. The tweets posted can broadly be categorized into two categories - Situational (having information that might be useful to the concerned authorities, such as the number of affected people in a particular region or contact information) or Non-situational (having information that might not be useful to the concerned authorities, such as sentiments like sympathy and personal opinions).

An example illustrating the situational versus non-situational tweet is given below:

Situational tweet: *Bomb blast in Hyderabad , 50 injured , say officials.*

Non-situational tweet: *Sitting in the path of a hurricane how fun.*

Hence, it is crucial to develop a classification model to help in classifying tweets as situational and non-situational. Another challenge is to deal with the large number of tweets that are posted during such events – this calls for summarization of the situational information.

In the current paper, we have proposed a classification and summarization framework, namely, *GenSumm* for generating an informative summary that can be used by disaster-monitoring agencies to provide immediate relief. The proposed

[1] https://www.omnicoreagency.com/twitter-statistics/
[2] https://www.businessofapps.com/data/twitter-statistics/

- *Diksha Bansal, Rahul Grover, and Sriparna Saha are with the Department of Computer Science and Engineering, Indian Institute of Technology, Patna, Bihar 801106, India. E-mail: {diksha.cs17, rahul.cs17, sriparna}@iitp.ac.in.*
- *Naveen Saini is with the Department of Information Technology, Indian Institute of Information Technology, Allahabad, Uttar Pradesh 211015, India. E-mail: nsaini1988@gmail.com.*

Manuscript received 12 January 2021; revised 12 July 2021; accepted 13 November 2021. Date of publication 30 November 2021; date of current version 18 November 2024.
The work of Dr. Sriparna Saha was supported in part by Young Faculty Research Fellowship (YFRF) Award, in part by the Visvesvaraya Ph.D. Scheme for Electronics and IT, and in part by the Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia) for carrying out this research.
(Corresponding author: Diksha Bansal.)
Recommended for acceptance by C. Clavel.
Digital Object Identifier no. 10.1109/TAFFC.2021.3131516

model is a two-stage process. In the first stage, a classification framework is proposed which simultaneously solves two tasks together namely, sentiment analysis and tweet classification, to identify situational tweets. In the second stage, various clustering algorithms and generative modelling are utilized for generating the extractive summary of the situational tweets identified from the first stage.

For classification task, we are proposing a multi-task deep learning model that takes embeddings from BERTweet [5] as features for the classification task. The model solves two tasks simultaneously: a) classify tweets as situational or non-situational b) predict the tweet's sentiment as positive or negative. Non-situational tweets are expected to have sorrow, anger, pain sentiments. Hence, the similarity between the two tasks is exploited to improve learning efficiency.

The language model BERT [6] (the Bidirectional Encoder Representations from Transformers) is efficient for solving various tasks and is the current state-of-the-art. BERTweet [5] is the first public large-scale language model pretrained for English Tweets based on the same model configuration as BERT-base[3]. BERTweet outperformed previous state-of-the-art models on three downstream Tweet NLP tasks: part-of-speech tagging, named entity recognition, and text classification.

Most of the extractive-summarization approaches [7], [8], [9], [10], [11], [12] cluster the dataset and then, extract tweets from each cluster. Clustering is performed to group similar sentences/tweets and reduce data redundancy. A single clustering algorithm cannot be relied on for all types of datasets, as every dataset has a unique distribution. Various clustering algorithms have been proposed till now to fit different datasets as discussed in Section 4.1. A common approach followed is to create an ensemble for various clustering algorithms [13], [14], [15]. In this paper, we have utilized different existing clustering algorithms and ensembled the solutions using Generative modelling [16].

Generative modelling is widely used to investigate data from weak supervisory sources, including knowledge bases, heuristic laws, noisy crowd labels, or even other classifiers. Weak supervisory sources often have limited accuracy and coverage. These labels are not regarded as gold labels because they can be noisy and conflicting. Generative modelling infers the dependence and correlation among various weak supervisory approaches to generate final labels. Recently, the researchers of Stanford University proposed a new standard of a generative model named *Snorkel*[4]. In the proposed approach, widely used existing clustering algorithms are considered as weak supervision sources. Then *Snorkel* is employed as a smart ensemble to produce the right consensus solution.

After partitioning the dataset, top-ranking tweets are extracted from each cluster to generate the summary. Tweets are ranked based on the sum of two features as described in more detail in Section 4.3.1.

For the classification task, we have experimented with numerous model designs and features to develop a generic classifier that can perform equally well for various disaster-events. Various features explored are sentence embeddings from BERTweet, emotional-aware embeddings [17], and sentiment labels of the tweets. Finally, multi-task learning is utilized for simultaneously solving two tasks, a) informative tweet classification; b) sentiment analysis, leveraging the similarities between both the tasks.

The proposed approach performed better than the state-of-the-art *COWTS* [3] and *MOOTweetSumm* [2] summarization models on four disaster-related events. Each event consists of human-annotated golden summaries for two breakpoints at 2000 and 5000 tweets. We have calculated the ROUGE-L F1-score for comparison purposes. The mean of ROUGE-L F1-score over all four datasets for both breakpoints attained by the proposed method has improved by 5.16% and 2.53% over *COWTS* and *MOOTweetSumm*, respectively.

The current work has the following advantages over prior studies:

- The paper reports about the development of a two-stage framework for classification and summarization of tweets. First stage involves identification of situational tweets and the second phase deals with the generation of summary based on the situational tweets.
- First stage of the proposed approach requires classification of tweets into two categories, namely situational and non-situational. The proposed classification model is based on the multi-task approach that solves two tasks: a) tweet information classification, b) sentiment analysis of tweets, simultaneously. Similarities in both the tasks are leveraged to improve learning efficiency.
- For summarization, various well-known clustering algorithms are utilized and then ensembled using generative modelling. The proposed summarization approach performs better than the state-of-the-art models on four datasets.
- The proposed framework can be easily employed for languages other than English. Both classification and summarization models utilize sentences embeddings from BERTweet [5]. BERT is known for its multi-lingual cross-learning properties. Hence, the proposed classification model can be used for cross-lingual transfer learning. Similarly, sentence embeddings generated using BERT can be used for the summarization task.

The rest of the paper is divided as follows: Section 2 presents existing works related to tweet classification and summarization. Section 3 discusses the proposed classification procedure. The proposed summarization approach is explained in Section 4. In Section 5, experimental setting is described. Results are presented in Section 6. In Section 7, a case study is presented. Finally, the paper is concluded in Section 8.

2 Related Work

In this section, we have discussed about the existing works.

2.1 Tweet's Classification in Disaster Event

Several attempts have been made to separate situational tweets from non-situational tweets. Bag-of-word model is employed in [18], [19], [20] for classification. Hence, the

[3] https://github.com/google-research/bert
[4] https://snorkel.org/

TABLE 1
List of Recent Works in the Field of Microblog Summarization

Method	Description
TSum4act [7]	This method first identifies informative tweets; then assigns informative tweets to topics; finally summarization is done.
PV-REL [28]	Proposed a method for summarising microblogs that takes into account the paragraph vector and semantic structure to alleviate feature sparsity.
MOOTweetSumm [2]	Various statistical quality measures capturing various aspects of summary, are optimised simultaneously using the search capability of a multiobjective differential evolution technique.
SOM + GSOM [8]	Self-organizing Map (SOM) [29] is used to reduce the available set of tweets to a smaller set, and then Granular Self-organizing Map (GSOM) [30] is used to extract relevant tweets.
TAKE [31]	On Twitter, a topic-based microblog summarization framework is used, taking into account both the time and the meaning of tweets.
COWTS [21]	A classification-summarization framework that is used to fragment the tweets to eliminate non-informative tweets from the summary.
MEDSUM [3]	Authors proposed a MEDical dictionary based tweet SUMmmarization approach which generates the summary for different stake-holders fulfilling different information needs during epidemic.
EnGraphSumm [32]	Proposed a graph-based unsupervised approach to produce better summaries by ensembling the summaries generated by various baseline algorithms.
Learn2Summ [32]	This approach is similar to *EnGraphSumm*. Only difference is that it is supervised approach instead of unsupervised.

model's learning efficiency is heavily dependent on the vocabulary of disaster events. Authors in [21] have developed a classification model based on domain-independent lexical features to distinguish among tweets. With the advent of deep learning models, researchers are employing neural networks for the classification task. Caragea et al. [22] have used a convolution neural network (CNN) [23] for classification of disaster-related tweets where tweets are represented using the bag-of-word model. Various features/representations are proposed to train deep-learning models. A language-agnostic model considering only Twitter-specific metadata of each tweet as base features is proposed in [24]. In [24], the datasets used for training purposes cover three types of natural disasters *Earthquake, Flood, and Fire*. An approach is proposed in [25], where convolutional neural network (CNN) is utilized for feature extraction, and artificial neural Nnetwork (ANN) is employed for classification. Authors in [25] have trained and tested the model on Hurricane Harvey, which hit Texas in the US on August 25, 2017. Alrashdi et al. [26] have developed a bidirectional long short-term memory (Bi-LSTM) [27] model which uses Glove[5] word embedding to represent the tweet.

Most of the proposed approaches are trained on datasets that cover only certain disaster events or based on lexical features that are unable to understand the context or sentiment of the tweet. We have curated several natural and human-made disaster events to create a generic dataset that can perform equally well on cross-domain datasets. The proposed deep-learning-based multi-task classifier utilizes the features extracted from BERTweet and leverages the similarities between the tasks of information classification and sentiment analysis.

2.2 Tweet Summarization

Numerous approaches have been proposed in the literature to accomplish the task of summarization. Table 1 gives an overview of some of the recent works. More detailed discussion is provided below. Authors in [7] have proposed *TSum4act* where first informative tweets are separated from noisy tweets and then divided into predefined classes (i.e., casualties, cautions, and donations). Clustering followed by the ranking of tweets are carried out for each predefined class of tweets. Top ranked tweets are extracted from each cluster as a summary. Authors in [33] have extracted informative sentences considering semantic and relation features, which are calculated using *Paragraph Vector*.

Multi-objective optimization, where various statistical quality measures reflecting summary quality are optimized simultaneously, was employed in [2]. A fusion of two architectures, *self-organizing map (SOM) and granular self-organizing map (GSOM)*, was presented in [8]. Here, the available set of tweets are first reduced to a smaller set using SOM, and then relevant tweets are extracted using GSOM. Authors in [34] have proposed *Time Aware Knowledge Extraction* methodology that relies on a temporal extension of Fuzzy Formal Concept Analysis and summary is extracted considering semantics and timestamps of the tweets. Authors in [35] have developed a microblog summarization approach for disaster-based events where informational tweets are filtered out using SVM classifier based on lexical features of tweets. Then the summary is formulated using an Integer Linear Programming (ILP)-based technique. Authors in [3] have presented a classification-summarization approach for summarizing tweets during disaster outbreaks. Here, tweets are classified into various disease-related categories: *Symptom, Prevention, Transmission, Treatment, Death report, and Non-disease.*, and different kinds of summaries are generated for affected and vulnerable communities and health organizations. Ensemble schemes that can combine the outputs of multiple base summarization algorithms are proposed in [36].

There also exist some works in the literature on opinion mining which can also be incorporated for the possible extension of our proposed approach. For example, in [37], an argumentation-based opinion mining framework is proposed which has focused on the analysis of online user-

[5] https://nlp.stanford.edu/projects/glove/

TABLE 2
Data (Tweets Containing Multiple Languages) Distribution of Informativeness Across Different Sources

Class ↓ Dataset →	CrisisLex	CrisisNLP	SWDM13	ISCRAM13	DRD	DSM	CrisisMMD	AIDR	Total
Situational	42,140	23,694	716	2,443	14,849	3,461	11,488	2,968	101,759
Not Situational	27,559	16,707	141	78	6,047	5,374	4,532	3,901	64,339
Total	**69,699**	**40,401**	**857**	**2,521**	**20,896**	**8,835**	**16,020**	**6,869**	**166,098**

generated content. It's a graph-based approach and can be utilized in our framework as a graph-based clustering technique followed by extraction of relevant tweets from each cluster to formulate a summary. Similarly, the authors of [38] have also investigated the three-phase process for opinion mining in computational advertising.

3 Classification Model for Tweets

Here, we have discussed the approaches followed for distinguishing situational tweets from non-situational tweets and datasets used for training purposes.

3.1 Problem Statement

People frequently use Twitter and other social media to post their opinions, feelings, emotions. A study[6] has shown that Twitter users with small local networks (with 100-200 followers) increase their activity more than those with more extensive networks in a disaster outbreak and are more likely to seek and exchange useful information in emergencies. Millions of tweets are posted with varying characteristics, including situational updates, views, sentiments, prayers, and public opinion. However, most of the tweets comprising of prayers, emotions are not useful for disaster-relief agencies. Tweets regarding situational updates, including information of displaced people, evacuations, missing or trapped or found people, donations or offers, rescue operations, needs, infrastructure or utilities damage, are crucial and need to be filtered out from the rest of the non-situational tweets.

Manually filtering such tweets can be cumbersome and time-consuming. An efficient system is required to filter situational tweets from the stream of the tweets. Given a tweet t, we need a method $\Psi : t \rightarrow \{0, 1\}$, that takes tweet t as input and returns label of the tweet t as 0 (non-situational tweet) or 1 (situational tweet).

3.2 Dataset Used for Classification

We have used CrisisBench dataset[7] [39] for the classification purposes. CrisisBench is obtained by consolidating eight publicly available datasets namely CrisisLex (CrisisLex26 [40], CrisisLex6 [41]), CrisisNLP [42], SWDM2013 [42], ISCRAM13 [43], Disaster Response Data (DRD)[8], Disasters on Social Media (DSM)[9], CrisisMMD [44], [45] and data from AIDR[10]. The number of situational and non-situational tweets are shown in Table 2. Note that the used dataset is already available in splitted form for training and testing; therefore, we have used them as it is.

3.3 Classification Approaches

We have designed four deep-learning based techniques for classifying the tweets. For this purpose, the sequential model from TensorFlow Keras framework[11] is utilized. Features and architecture of the model, as shown in Fig. 1 are discussed below:

3.3.1 *BERTweetClassifier*

Sentence Embeddings from BERTweet [5] are used as features. Note that BERT outputs a vector size of 768 as a sentence embedding. A dense[12] layer is applied on the top of embeddings. A dense layer is a regular densely-connected neural network layer. First, dot product is calculated between input *input* and the weight matrix W in our dense layer. Then, a bias vector *bias* is added to it. At last, activation *Activation* of the output values for each unit in the output layer is performed as shown in the equation below:

$$Output = Activation(dot(input, W) + bias) \tag{1}$$

Here, softmax[13] function is used as the activation function in the dense layer. The output layer contains two nodes describing tweets' probability of belonging to class {0,1}, respectively. Finally, the tweet is assigned a label for the class that has the highest probability.

Model was trained for 20 epochs using Adam optimizer[14] with early stopping[15] monitoring validation loss. Patience value was set to 3. To counter class imbalance problem, Focal loss [16] is used as proposed in the RetinaNet paper [46]. It is extremely useful for classification when we have highly imbalanced classes. It focuses on hard examples by down voting well-classified examples and hence loss value is higher for a sample which is misclassified by the classifier.

3.3.2 *Emotion Aware Classifier*

BERT applies the Masked Language Model (MLM) to use the left and the right context during pre-training to create deep bidirectional Transformers. Hence, the words that occur in the same context tend to be semantically closer.

[6] https://www.sciencedaily.com/releases/2019/02/190213142654.htm

[7] https://crisisnlp.qcri.org/crisis_datasets_benchmarks.html

[8] https://appen.com/open-source-datasets/

[9] https://data.world/crowdflower/disasters-on-social-media

[10] http://aidr.qcri.org/

[11] https://www.tensorflow.org/guide/keras

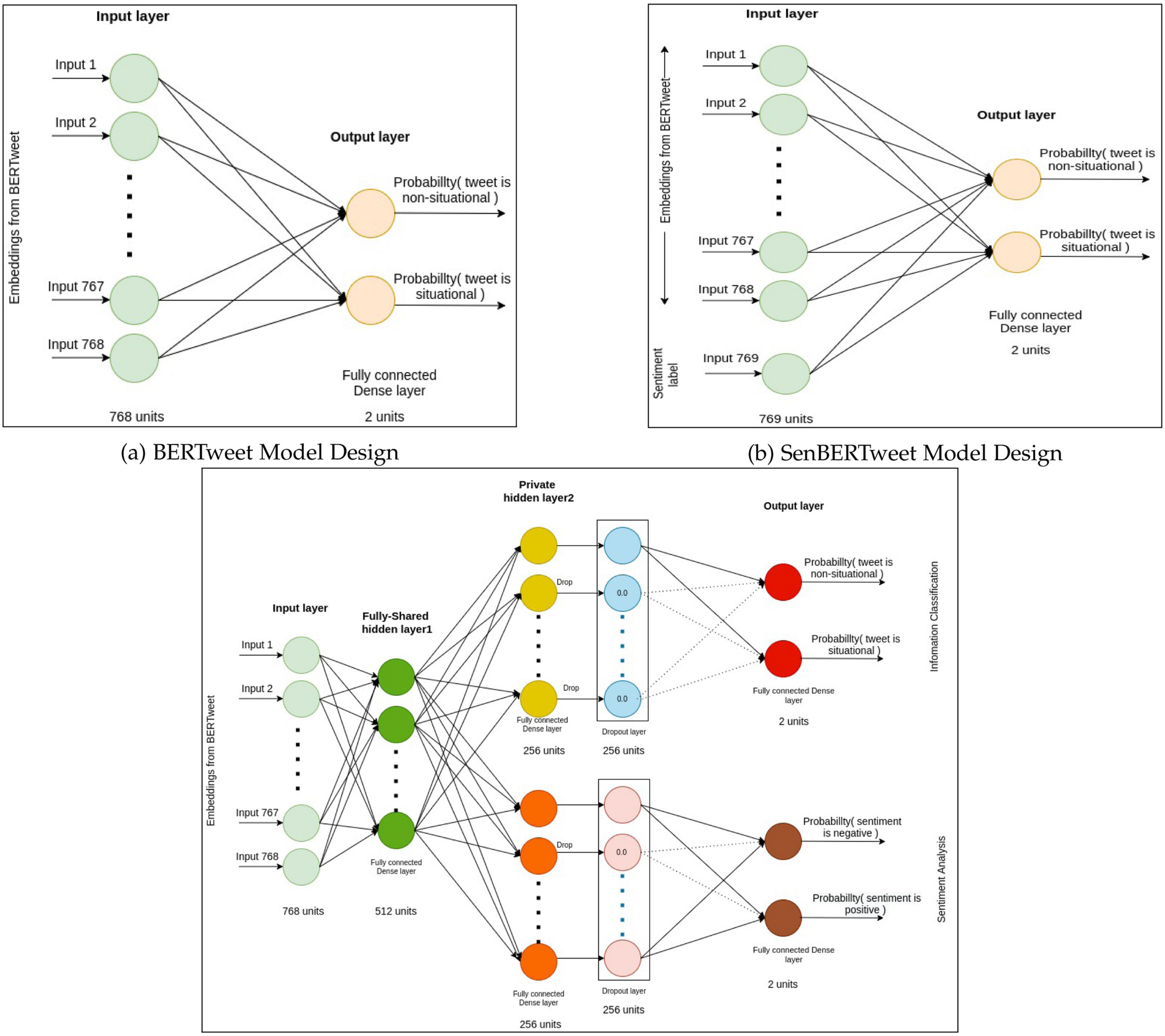
[12] https://keras.io/api/layers/core_layers/dense/

[13] https://keras.io/api/layers/activations/#softmax-function

[14] https://keras.io/api/optimizers/adam/

[15] https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

[16] https://www.tensorflow.org/addons/api_docs/python/tfa/losses/SigmoidFocalCrossEntropy


Input layer
Input 1
Input 2
Input 767
Input 768
Embeddings from BERTweet
Output layer
Probabillty(tweet is non-situational)
Probabillty(tweet is situational)
Fully connected Dense layer
768 units
2 units
Input layer
Input 1
Input 2
Input 767
Input 768
Input 769
Embeddings from BERTweet
Sentiment label
Output layer
Probabillty(tweet is non-situational)
Probabillty(tweet is situational)
Fully connected Dense layer
2 units
769 units
Private hidden layer2
Output layer
Input layer
Fully-Shared hidden layer1
Input 1
Input 2
Input 767
Input 768
Embeddings from BERTweet
Drop
Drop
0.0
0.0
Probabillty(tweet is non-situational)
Probabillty(tweet is situational)
Infomation Classification
Fully connected Dense layer
Dropout layer
Fully connected Dense layer
256 units
256 units
2 units
Fully connected Dense layer
768 units
512 units
Drop
0.0
Probabillty(sentiment is negative)
Probabillty(sentiment is positive)
Sentiment Analysis
Fully connected Dense layer
2 units
Fully connected Dense layer
Dropout layer
256 units
256 units

(a) BERTweet Model Design

(b) SenBERTweet Model Design

(c) Multitask Model Design

Fig. 1. Design of multi-task deep-learning based classification models used in our proposed approach.

Dissimilar words like *happy* and *sad* mostly occur in the same context. Such words can be interchanged easily without breaking grammatical rules. Hence, this complication leads to a rather undesirable outcome in predictive tasks related to sentiment or emotional state, as shown in [17]. We have utilized emotion-enriched word representations[17] which are trained by authors in [17] leveraging distant supervision and neural networks. Sentence embeddings are obtained by computing the average of word vectors of all the words in the tweet along each dimension, as suggested in the paper. Note that, we have removed the words that are not present in the vocabulary of the emotion-enriched word representations. This sentence representation is input to the classification model. Model design and experimental settings are the same as done in *BERTweetClassifier*, the difference being a) the way embeddings are calculated b) the size of sentence embedding is 300 in this case.

3.3.3 Sen-BERTweet

In *EmotionAwareClassifier*, embeddings for some tweets were not present as all the words were out of the vocabulary. To counter this problem, the sentiment label of the tweet is used as a feature along with BERTweet sentence embeddings. Here also, a dense layer is applied on the top of features similar to *BERTweetClassifier* described in Section 3.3.1.

Two weak classifiers, namely VADER [47] and TextBlob[18] are employed to generate sentiment labels for each of the instances in the dataset.

[17] https://www.dropbox.com/s/5egqnbktbfxp2im/ewe_uni.txt.zip?dl=0

[18] https://textblob.readthedocs.io/en/dev/quickstart.html

VADER (Valence Aware Dictionary for Sentiment Reasoning) is based on lexicons of sentiment-related words and is employed for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotions. On the other hand, TextBlob is a simple python library for Natural Language Processing (NLP) that supports complex analysis and textual data operations. TextBlob has semantic labels that help with fine-grained analysis, and it returns the polarity of a sentence. Polarity lies between [-1,1], -1 defines a negative sentiment and 1 defines a positive sentiment. There can be a third type of sentiment namely, 'neutral'; therefore, we have investigated the effect of using only two and three sentiment labels separately as discussed below.

- Two-way classification: Tweets are classified into positive or negative sentiments. The average of compound polarity returned by TextBlob and Vader is considered to determine the sentiment of the tweet. We have considered a tweet *t* belonging to positive sentiment if the averaged polarity for tweet $t \geq 0$; otherwise tweet *t* is considered to have a negative sentiment.
- Three-way classification: Tweets are classified into positive, negative, or neutral sentiments. Similar to two-way classification, here also, the average of compound polarity returned by TextBlob and Vader is considered to determine the sentiment of the tweet. A tweet is considered to have positive sentiment if the average polarity is greater than 0, neutral if average polarity equals to 0, and negative if the average polarity is less than 0.

Hence, a total of 769 (768 from BERTweet Sentence Embeddings, one from sentiment label) features are given as input to the model. The dense layer is employed to 769 features, and it outputs two neurons representing the probability of a tweet belonging to each class. All experimental settings are similar to *BERTweetClassifier* as described in the Section 3.3.1.

3.3.4 Multitask Classifier

Multi-tasked deep learning (MTL) has been extensively utilized in literature [48], [49], [50], [51], [52] to solve numerous tasks concurrently by leveraging similarities between different tasks. MTL can help in improving learning efficiency and can act as a regularizer. People often apply the knowledge learned from previous tasks to help them learn a new task. Like human learning, it is useful for multiple learning tasks to be learned jointly since other tasks can leverage the knowledge contained in a task. MTL aims to increase learning performance by forcing the model to learn a more generalized representation of the input data's features or attributes as it learns (updates its weights), not just for one specific task.

Motivated by the wide use of multi-task deep learning models for solving classification problems, we have employed MTL to simultaneously solve two different tasks: a) Situational versus non-situational tweet classification, b) Sentiment analysis of the tweet.

For training purposes, we have used sentiment labels for both the two-way and three-way classifications obtained from two weak supervision sources (VADER, TextBlob) as described in Section 3.3.3.

Given a tweet *t*, we have developed a model

$$\Psi : t \rightarrow \begin{cases} \text{Information Label} \begin{cases} 0, \text{ if t is non-situational tweet} \\ 1, \text{ if t is situational tweet} \end{cases} \\ \text{Sentiment Label} \begin{cases} \text{Two-way sentiment classification} \\ \begin{cases} 0, \text{ if t has negative sentiment} \\ 1, \text{ if t has positive sentiment} \end{cases} \\ \text{Three-way sentiment classification} \\ \begin{cases} 0, \text{ if t has negative sentiment} \\ 1, \text{ if t has neutral sentiment} \\ 2, \text{ if t has positive sentiment} \end{cases} \end{cases} \end{cases} \tag{2}$$

that takes tweet *t* as input and returns information label and sentiment label of the tweet *t* as output.

The designed model/architecture, employed to train the multi-task deep learning model is shown in Fig. 1c. We have used sentence embeddings from *BERTweet* of size 768 as input features. Our proposed model has used a common hidden layer (fully shared layer) for both tasks and task-specific layers (private layers) towards the end of the model. This technique is widely employed to reduce over-fitting risk by learning a representation for various tasks by a common hidden layer. We have also used a fully-connected dense layer (*color coded green*) with ReLU [53] activation function on the top of the sentence embeddings to decrease the input layer's size to 512. This layer acts as a common fully-shared layer for both the tasks. This layer is further reduced to two 256 units layers (*color coded yellow and orange*) using the dense layer with ReLU activation function as task-specific private layers for both tasks. In the output layer (*color coded red and brown*), the task-specific private layers are reduced to 2 units using dense layer with Softmax activation function and a dropout [54] of 50% for both tasks. The dropout layer randomly sets input units to 0.0 with a frequency of the given rate at each step during training time to avoid over-fitting. Finally, output units in each task represent the probability values of a tweet belonging to that particular class.

We have used Adam optimizer with default parameters for 20 epochs with early stopping monitoring validation loss to tune the model. Patience value was set to 3. The loss function used for both the tasks is focal loss. The final loss optimized by model is average of the Focal Loss generated by both the tasks. Mathematically focal loss can be defined as:

$$FL(p) = \begin{cases} -(1-p)^{\gamma} log(p), & \text{if } y = 1 \\ -p^{\gamma} log(1-p), & \text{otherwise} \end{cases}$$

In the above equation, $y \in \{\pm 1\}$ specifies the ground-truth class and $p \in [0, 1]$ is the model's estimated probability for the class with label y = 1. The focal loss is visualized for several values of $\gamma \in [0, 5]$.

The loss function optimized by the model is defined as below:

$$Loss = \frac{FL(p)_{info} + FL(p)_{sent}}{2} \tag{3}$$

TABLE 3
Proposed Multi-Tasked Classifier's Design Summary

Parameters	MultitaskClassifier	Description
Tasks	Information Classification Sentiment Analysis	Classify tweet as situational (1) or non-situational (0) Identify sentiment of tweet as positive (1) or negative (1)
Input features	BERTweet Sentence Embeddings	Vector of size 768
Shared layer	Dense layer (512 oupout units, 768 input units)	Connected to input features, ReLU activation used
Task specific layers	Dense layer (256 oupout units, 768 input units) Dropout layer of 50% Output Dense layer (2 output units, 256 input units)	Connected to shared layer, ReLU activation used Connected to task-specific dense layer Connected to task-specific dropout layer, Softmax activation used
Loss function	Sigmoid Focal Loss	Average of loss generated by tasks is optimized
Epochs	20	Early stopping monitoring validation loss, patience value set to 3
Optimizer	Adam	Default parameters used

where, $FL(p)_{info}$ and $FL(p)_{sent}$ are focal loss generated for information classification task and sentiment analysis task, respectively. A brief summary of the tasks, model design, parameters used are described in Table 3.

3.4 Comparative Methods

We have utilized convolution neural network (CNN), fastText and pre-trained transformer models as presented in [39] for comparison purposes. A very small improvement is observed in *BERTweet Classifier* and *Emotion Aware Classifier* as compared to *BERT* model. Brief descriptions of these models are given below:

- CNN: This architecture is used in the current state-of-the-art disaster classification model. The architecture followed is proposed in [55].
- fastText: Pretrained embeddings trained on Common Crawl are utilized, which are released by fastText for English [56].
- Transformer-based models: Pre-trained models have attained state-of-the-art performance on natural language processing tasks, and they've been used as feature extractors to solve downstream tasks like question answering and sentiment analysis and their rich contextualised embeddings might be useful in the disaster domain. Therefore, for our classification purpose, we have used BERT [57], DistilBERT [58], and Roberta [59].

3.5 Performance of Classification Techniques

In Tables 4 and 5, results attained by the developed classifiers for information classification and sentiment classification tasks are shown in terms of weighted F1-score values, respectively. We have performed five re-runs for each experiment, and calculated the average of the results. We have trained *BERTweet Classifier* and *EmotionAware Classifier* for sentiment classification task for comparison purposes. Model architecture and parameters are same as that of information classification task.

From these tables, a small decrease in F1-Score is observed using *Emotion Aware Classifier* as compared to the *BERTweet Classifier* approach for the information classification task but an increase is observed for the sentiment analysis task. This is due to the major drawbacks of the *Emotion Aware Classifier* which are: a) many tweets have to be dropped due to null embeddings as not a single word in the tweet was in the vocabulary; b) using the mean of word embeddings to generate sentence embedding does not consider the interactions between the words in a sentence, nor the word order. Hence, to check if both tasks are related or not, we employed *SenBERTweet Classifier* where we have used the sentiment as a feature.

Improved performance of 0.64% and 0.54% in F1-Score on two-way and three-way sentiment classifications, respectively, of the SenBERTweet approach compared to the BERTweet model illustrates that sentiment is a noteworthy feature in classifying tweets. This has motivated us to explore multi-task-based deep-learning models for two tasks: a) information classification; b) sentiment classification, which has further improved learning efficiency. An increase in F1-Score of 1.43% and 1.22% for two-way and three-way sentiment classification tasks, respectively, using *MultiTask Classifier* compared to *BERTweet Classifier* has proved the hypothesis that both tasks are related and similarities between both the tasks can be leveraged to enhance learning efficiency.

3.6 Analysis of Results of Various Approaches

In this section, we have presented error analysis for various classification approaches. In Table 6, we have shown the

TABLE 4
Classification Results (Weighted-F1) for Information Classification Task Using CNN, Fasttext (FT) and Transformer Based Models

Approach →	CNN	FT	BERT	D-B	RT	BT	EAC	SBT-2	MC-2	SBT-3	MC-3
weighted-F1	0.866	0.844	0.872	0.870	0.883	0.877	0.873	0.880	**0.888**	0.879	0.886

D-B: DistilBERT, RT: RoBERTa, BT: BERTweetClassifier, EAC: EmotionAwareClassifier, SBT: SenBERTweet, MC: MultitaskClassifier. Here, 2 and 3 refer to two-way classification and three-way classification for sentiment analysis, respectively.

TABLE 5
Classification Results (Weighted-F1) for Sentiment Classification Task Using BT: BERTweetClassifier, EAC: EmotionAwareClassifier, SBT: SenBERTweet, MC: MultitaskClassifier

Approach	BT-2	EAC-2	MC-2	BT-3	EAC-3	MC-3
Weighted F1-Score	0.714	0.721	**0.734**	0.602	0.623	0.642

2 and 3 refer to two-way classification and three-way classification for sentiment analysis, respectively.

TABLE 6
Confusion Matrix for Various Classification Approaches on Information Classification: BT: BERTweetClassifier, SBT: SenBERTweet, MC: MultitaskClassifier

Approach	BT		SBT-3		SBT-2		MC-3		MC-2	
Actual Labels ↓ Predicted Labels →	0	1	0	1	0	1	0	1	0	1
0	7071	717	6846	942	6661	1127	6660	1128	6550	1238
1	2259	11055	1660	11654	1388	11916	1286	12028	1085	12229

Here, 2 and 3 refer to two-way classification and three-way classification, respectively, for sentiment analysis.

confusion matrices[19] generated by various classification approaches on the testing data of CrisisBench Dataset. In a confusion matrix *C*, $C_{i,j}$ equals to the number of observations known to be in group *i* and predicted to be in group *j*.

The number of misclassified situational tweets are $2259, 1660, 1388, 1286, 1085$ (decreasing) and the number of misclassified non-situational tweets are $717, 942, 1127, 1128, 1238$ (increasing) in *BERTweet Classifier, SenBERTweet-3, MultiTaskClassifier-3, SenBERTweet-2, MultiTaskClassifier-2* classification approaches, respectively. Here, 2 and 3 refer to 2-way and three-way classification, respectively. A trade-off can be observed between misclassified situational tweets and non-situational tweets among various approaches commonly known as a precision-recall trade-off. Some of the examples are shown in Table 7. Most of the situational tweets are classified as non-situational tweets by *BERTweet Classifier* but correctly by *SenBERTweet-2, MultiTaskClassifier-2*.

It is worth noting that misclassification of a situational tweet as a non-situational tweet is more vulnerable as it might lead to missing some critical updates in the summary. Therefore, we have utilized *two-way sentiment classification MultiTaskClassifier* for information classification purposes. Hence, we have more misclassified non-situational tweets in the summarization process.

There are various versions of $RoBERTa$ model and in this paper, we have used its four versions namely, $RoBERTa_{large}$, $RoBERTa_{base}$, $XLM-R_{large}$, $XLM-R_{base}$, for comparison purpose. On comparing $RoBERTa$ with BERTweet, $RoBERTa_{large}$ based model has outperformed BERTweet classifier by 0.93%. A similar behaviour is obtained in the BERTweet paper [5] where BERTweet does better than its competitors, $RoBERTa_{base}$ and $XLM-R_{base}$, but, obtains lower performance as compared to $RoBERTa_{large}$ and $XLM-R_{large}$ models. Note that in Tables 4 and 5, the reported results for $RoBERTa$ are corresponding to its version $RoBERTa_{large}$ as it performs better than other versions. On comparing all classifiers, our *MultitaskClassifier*-based approach is able to outperform $RoBERTa_{large}$ by 0.29% on an average.

Noted that tweet sentiment analysis is a challenging task as tweet may comprise of expressions namely abbreviations, acronyms, misspelled words and slang words. Sarcasm detection, thwarted expressions, use of code-mixed and code-switched texts pose challenges for traditional sentiment classifiers. We have observed such examples where VADER and TextBlob are not able to predict the sentiments of the tweets correctly. Some examples are shown in Table 9. Three-way classification is supposed to outperform two-way classification but two-way classification for multitask approach has performed 0.21% better as compared to that of three-way classification. We believe if sentiment labels are manually labelled, then three-way classification would have performed better as a lot of neutral tweets are labelled as positive or negative by TextBlob and VADER sentiment classifiers.

3.7 Statistical Significance of Classification Results

We have also checked the statistical significance of the classification results at 5% significance level. Results for statistical t-tests are shown in Table 8. It has been observed that p-values provided by this test are less than 5% which validates that *MultitaskClassifier* approach is statistically significant than other comparative approaches.

4 Tweet Summarization

In this section, we have presented the algorithm used for tweet summarization. Most of the summarization algorithms are based on clustering the tweets and then ranking different clusters to extract top tweets. Clustering solutions are highly dependent on distribution of data and thus effect the final performance. Here in the proposed approach named GenSumm, we ensemble various cluster solutions to generate a single best solution.

The proposed GenSumm approach can be divided into three major steps:

[19] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

TABLE 7
Predicted Examples on the Test Data for Information Classification Task Using BT: BERTweetClassifier, EAC: EmotionAwareClassifier, SBT-2: SenBERTweet (Two-Way Sentiment Classification), MC-2: MultitaskClassifierClassification Results (weighted-F1) for Sentiment Classification Task Using BT: BERTweetClassifier, EAC: Emo-tionAwareClassifier, SBT-2: SenBERTweet

Tweet	BT	SBT-2	MC-2	Actual Label
Death toll in Philippines quake jumps to 93 http://t.co/ZiykJpZxbe via @USATODAY	0	1	1	1
Most of them or 51% were diagnosed as having tinea pedis, followed by dermatitis at 20%, respiratory diseases at 18% and wounds from accidents at 7%.	0	1	1	1
RT @BBCBreaking: Another 2 bodies found after oil train explosion in eastern Canada	0	1	1	1
RT @WHO: A person who is infected w/ #Ebola is only contagious after s/he has started to have symptoms	0	1	1	1
The world really is ending now there is an explosion in Texas	0	1	1	1
RT @RajuRaina1: @JasumatiPatel @DR_M_DUTT FLOOD IS NOT ONLY IN KASHMIR AS SHOWN BY MEDIA JAMMU HAD SUFFERED MORE LOSE AS HUMAN N PROPERTY	0	1	1	1
My mom just told me 43 people died already from Sandy ..	0	1	1	1
THE NATIONAL WEATHER SERVICE IN LITTLE ROCK HAS ISSUED A * SEVERE THUNDERSTORM WARNING FOR... VAN BUREN COUNTY IN _ http://t.co/KJsvW06GBV	0	1	1	1
It's getting to be hazardous getting into this world alive. https://t.co/BJZSSw4tid	1	0	0	0
@Anne_R_u_Ok: 6.5-magnitude quake strikes off Guatemala http://t.co/mtckd1cN via @addthis	0	1	1	1
#BalochGenocide: Five more dead bodies found from Panjgur were brutally killed by Pakistani forces. #Balochistan @UN https://t.co/fjGcwHm82o	0	1	1	1
Please have more officials in the street.	1	0	0	1
I know the situation is still very serious but does anyone else feel better now that it has stopped raining? #yycflood	0	1	1	0
The hurricane might ruin my concert plans for #collidewiththeskytour @bandPr0blems	1	1	0	1
Lets all find a way to help those in need in #Colorado	1	0	0	1
RT @calgarysun: RCMP confirm two bodies pulled from Highwood River one woman remains missing #ABFlood	0	1	1	1
RT @wildfiretoday: I-70 is now CLOSED from exit 49 to exit 62. Alternate route is the Debeque Cutoff to Hwy 65 #COfire #pineridgefire #gjco	0	1	1	1
In Bonrepos we don't know how to get that Food pass. How can we get it?	1	0	1	0
My mom is watching a show about bridges breaking/falling and the people on them drowning in their cars aka one of my biggest fears ????	0	1	0	0
We can go get food clothing or hygiene products for a couple families	1	0	1	1
I knew those pints would come in handy for something #yycflood @fiascogelato http://t.co/yCbLaoUn85	1	0	0	1

0 suggests that tweet is non-situational and 1 represents that tweet is situational.

- Clustering: This step is responsible for obtaining various clustering solutions. It is described in Section 4.2 in detail.
- Generative Modelling: This step is responsible for ensembling multiple clustering solutions using generative modelling. We have discussed the steps involved in detail in Section 4.2.
- Summary Generation: This step is responsible for extracting tweets from final cluster solution to generate summary. Steps required are explained in Section 4.3.

All the above steps are represented in Fig. 2.

4.1 Clustering Techniques

We have employed widely used clustering algorithms mentioned below:

- DBSCAN[20]: Density-Based Spatial Clustering of Applications with Noise is a density-based clustering algorithm. It assumes that clusters are dense regions and are partitioned by regions of lower density. DBSCAN works well for arbitrary shaped clusters and is capable of detecting outliers. It automatically determines the number of clusters. It takes two parameters a) *epsilon*: The distance that specifies the neighborhoods. Two points are considered neighbors if the distance between them is less than or equal to *epsilon*; b) *min_samples*: It refers to the minimum number of data points to define a cluster. In some cases, determining both parameters might be challenging and requires domain knowledge. It is not befitted if clusters are very different in terms of in-cluster densities.
- K-Means[21]: K-means clustering takes a parameter k and identifies k number of centroids. Every data point is allocated to the nearest cluster. Hence, it tries to minimize distances within a cluster and maximize the distances between different clusters. The approach

[20] https://scikit-learn.org/stable/modules/clustering.html#dbscan

[21] https://scikit-learn.org/stable/modules/clustering.html#k-means

TABLE 8
Statistical T-Test for Various Approaches With MC-2 Approach as Compared to CNN, Fasttext (FT) and Transformer Based Models

Approach	CNN	FT	BERT	D-B	RT	BT	EAC	SBT-2	SBT-3	MC-3
t-value	-6.6422	-12.8475	-5.4325	-5.7886	-2.5876	-5.8469	-4.4434	-2.8764	-4.2096	-0.2462
p-value	0	0	0	0	0.0098	0	0	0.0041	0	0.8056
statistically significant?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

D-B: DistilBERT, RT: RoBERTa, BT: BERTweetClassifier, EAC: EmotionAwareClassifier, SBT: SenBERTweet, MC: MultitaskClassifier. Here, 2 and 3 refer to two-way classification and three-way classification for sentiment analysis, respectively.

TABLE 9
Examples of Misclassified Tweets for Sentiment Analysis Task

Tweet	Polarity	Expected Label	Label using TextBlob and Vader
Hello concerning the food distribution for 3500 people suffering in delmas 33 on the football field	-0.2384	Neutral	Negative
Sitting in the path of a hurricane how fun	0.4053	Negative	Positive
@acewzrd_: I'm not ok as in right now, I may seem like it but I'm not at all.	0.0483	Positive	Negative
RT @TIME: Typhoon #Bopha: About 350 killed, 400 missing in the Philippines ∣ http://t.co/5v77sPtc (via @TIMEWorld)	-0.4859	Neutral	Negative
It was predicted that the said Typhoon Pablo is more destructive than Sendong. But we have The Almighty God who'll spare us from harm.	-0.3727	Positive	Negative
Calgary Saddledome flooding leads to cancellation of some Stampede events http://t.co/UngDuBN2Pu - @CBCNews	-0.2108	Neutral	Negative
Out mayor is the best! Haha love him #yycflood #yyc	0.7400	Negative	Positive
I really wish they'd build tornado shelter in their house. No basements there. After all they live in Norman smack dab in tornado alley!	0.2275	Negative	Positive
It amazes me,the way diseases are being termed after a particular region.... Middle East Respiratory Syndrome..... O ma ga oo	0.3303	Negative	Positive
1/3 Rain water of housing society n other buildings shd be poured in bourwells. Kindly spread msg to avoid flood @indianexpress @rezhasan	0.2438	Neutral	Positive
Response to the Boston Marathon Tragedy http://t.co/N1ATYMZ5Nu	-0.3299	Neutral	Negative
RT @YourAnonNews: BREAKING UPDATE: At least 51 people confirmed dead after Oklahoma tornado - Medical Examiner's office http://t.co/7iHq1rc. . .	-0.3368	Neutral	Negative
The road, which was closed for vehicular movement between Jogiroad - Nagar (Km 208 - 274) on 25th July, 2004 has been reopened for vehicles upto 16.2 tonnes.	-0.0500	Neutral	Negative
The 2,311 flood victims are receiving assistance from the township, and a 24-hour flood watch has been set up.	-0.1591	Neutral	Negative
2010 now I am sleeping on the streets. Please help me. I am suffering a lot. i would like to find shelter.	0.2634	Negative	Positive

Polarity refers to the average polarity returned by TextBlob and Vader.

can draw only linear boundaries and is sensitive to outliers.

- Agglomerative Hierarchical clustering[22]: It generates the clustering structure in a hiearchical manner. It generates a tree like structure which is called dendogram. Here, every point/observation is considered as a cluster at first. A pair of clusters which are closest to each other are merged to build a hierarchy of clusters. Hence, the approach creates a tree of clusters by iteratively grouping or separating data points. The algorithm works very slowly in the case of large datasets.
- Spectral Clustering[23]: Points/observations are considered as graph space and mapped to a lower-dimensional space. Hence clustering problem is reduced to graph-partitioning where nodes are segregated to form clusters. The algorithm can perform well with a wide variety of shapes of data as clusters. Clusters are not assumed to be of any specific shape/distribution. However, it requires the number of clusters to be determined beforehand and can be costly to compute.
- Birch Clustering[24]: Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) summarizes large datasets into smaller, dense regions that retain most of the information. The approach uses a tree structure to create a cluster and can cluster incrementally and dynamically to produce the best quality clustering for a given set of resources. BIRCH uses a multi clustering technique, wherein a basic

[22] https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering

[23] https://scikit-learn.org/stable/modules/clustering.html#spectral-clustering

[24] https://scikit-learn.org/stable/modules/clustering.html#birch

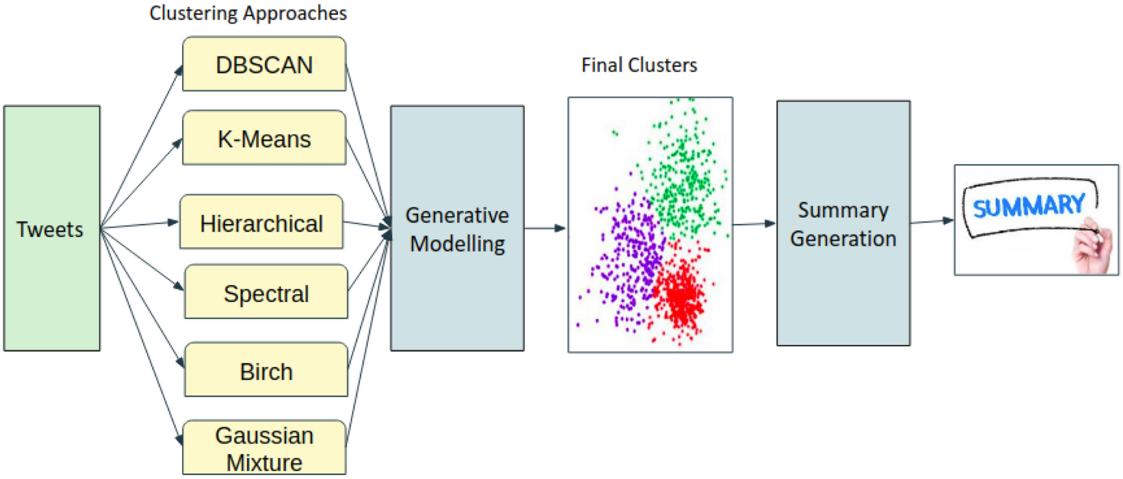


Fig. 2. Multitasking neural network architecture.

and good clustering is produced as a result of the first scan, and additional scans can be utilized to further improve the quality of clustering. It has three hyper-parameters (number of clusters, threshold, and branching factor) that make the use of the algorithm challenging.

- Guassian Mixture[25]: Gaussian Mixture is a probabilistic model assuming a certain number of Gaussian distributions are present, and each of these distributions represents a cluster. It uses a soft clustering approach for distributing the points in different clusters. It is the fastest algorithm for learning mixture models. It will not bias the means towards zero or bias the cluster sizes to have specific structures that might or might not apply as this algorithm maximizes only the likelihood. However, the algorithm may diverge and find solutions with an infinite likelihood in the case of insufficiently many points per mixture.

4.2 Generative Modelling

We have discussed the procedure of ensembling the cluster solutions in this section.

We have modeled the clustering problem as a classification task where every observation is labeled from 0 to $num\_clusters - 1$ where *num_clusters* is the total number of clusters formed by the DBSCAN approach. In simple words, *num_clusters* number of classes are created where each class has labels from 0 to $num\_clusters - 1$. We define label vector for a particular solution as a list of labels generated. For instance, consider 5 points *A,B,C,D,E* where *A, D* points belong to the first cluster and rest *B, C, E* points belong to second cluster. Here label vector can be {1,2,2,1,2} or {2,1,1,2,1}.

Thus, each clustering approach listed in Section 4.1 generates label vector for the given dataset. These approaches are referred to as weak supervision sources. The label vectors generated by weak supervision sources might be conflicting, noisy, and inaccurate. Thus, we ensemble the generated label vectors using generative modelling. More precisely, we have used Snorkel [60] that is widely used to combine labels from weak supervision sources. It takes various clustering solutions (weak supervision solutions) as inputs, discovers the dependencies and correlations among multiple solutions in the input, and infers a probabilistic solution. The critical detail here is that the generative model is built without any access to the ground truth or true labels.

Before passing label vectors as inputs to *Snorkel*, we have to ensure consistency between labels of different cluster solutions. Two cluster solutions can be the same, but their labels might be permuted. For instance, label vector X for one solution can be {112233444} and label vector Y for another solution can be {441133222} though both solutions represent the same partitioning. Hence, we have reorganized the label vector before ensembling. Suppose we have a reference clustering label vector X, to which we have to map a clustering label vector Y. For each cluster label *i* in X, we find the points that belong to cluster *i* as per solution X. Then we find what are cluster labels for these points as per solution Y. Since these points may have multiple cluster labels as per Y, we find the cluster label where most of these points are assigned as per Y. If this cluster label is *j*, we relabel all the points which have cluster label *j* in Y to cluster label *i*. It is repeated for each class label in X. Detailed idea is described in [61].

4.2.1 Snorkel Setup

In this section, we have discussed the implementation details of the proposed generative modelling.

Our goal is to develop a generative model p_θ that predicts label $y \cup Y$ for given data point $\chi \in X$ by integrating the weak supervision labeling functions obtained from clustering approaches listed in Section 4.1. Here X is the set of data points and Y refers to the number of classes or basically number of clusters *num_clusters*. Labeling functions (LFs) are programmatic rules and heuristics used in *Snorkel* to express various weak supervision sources such as patterns, heuristics, external knowledge basis, and more. LFs might be noisy, conflicting, or uncorrelated to each other. LFs take the data point as input and either assign a label to it (in our case, the class label) or abstain (-1 if the data point belongs to noise). Hence, LFs can be considered as black-box

[25] https://scikit-learn.org/stable/modules/mixture.html#mixture

functions, λ: $X \rightarrow Y \cup \{\emptyset\}$ that take data points as input and return output label. Here $\emptyset$ is used to denote that some of the labels may abstain. In our case, we have six LFs: λ_{dbscan}, λ_{kmeans}, $\lambda_{aggomolerative}$, $\lambda_{spectral}$, λ_{birch}, $\lambda_{guassian}$. Each LFs takes the label vector (after reorganization) generated from the corresponding clustering approach as an external knowledge base and returns the input data point label.

Assuming m data points in X and $n = 6$ labelling functions, *Snorkel* produces a matrix of LFs' outputs $\Lambda = Y \cup \{\emptyset\}^{mXn}$, where $\Lambda_{ij} = \lambda_j(x_i)$, by applying LFs over unlabelled data points in X that might contain overlapping and conflicting labels. *Snorkel* infers the dependencies and correlations between various LFs' outputs to predict probabilistic final labels $\tilde{Y} = (\tilde{y_1}, \tilde{y_2}, ..., \tilde{y_{m-1}}, \tilde{y_m})$ where $\tilde{y_i} \in Y$.

Snorkel improves the predictive performance by modelling statistical dependencies by focusing on pairwise correlations among various LFs. Snorkel builds a factor graph to develop the generative model and employs structure learning to select C set of labeling function pairs *(i,j)* as correlated. Three different factor types are used to encode the generative model $p_\theta(\Lambda, Y)$ described below:

- Labeling propensity:

$$\varphi_{i,j}^{Lab}(\Lambda, Y) = 1\{\Lambda_{ij} \neq \phi\} \tag{4}$$

- Accuracy:

$$\varphi_{i,j}^{Lab}(\Lambda, Y) = 1\{\Lambda_{ij} = y_i\} \tag{5}$$

- Pairwise correlations:

$$\varphi_{i,j,k}^{Lab}(\Lambda, Y) = 1\{\Lambda_{ij} = \Lambda_{ik}\} \qquad (j,k) \in C \tag{6}$$

Snorkel defines a concatenated vector of these factors for all LFs (in our case, six LFs) and potential correlations C as $\phi_i(\Lambda, Y)$ along with the corresponding vector of parameters, $\theta \in \mathbb{R}^{(2n+C)}$.

Hence, the model is defined as described in the equation below:

$$p_\theta(\Lambda, Y) = \xi^{-1}\exp\left[\sum_{i=1}^{m} \theta^T \varphi(\Lambda, y_i)\right] \tag{7}$$

where ξ is the normalizing constant.

Snorkel learns the model without any ground truth labels. To achieve this, it aims to minimize the negative log marginal likelihood as described in the equation below:

$$\tilde{\theta} = \arg\min_{\theta} -\log \sum_{Y} p_\theta(\Lambda, Y) \tag{8}$$

This objective is optimized by interleaving stochastic gradient descent steps with Gibbs sampling ones. Then, probabilistic training labels or predictions are defined as

$$\tilde{Y} = p_\theta(Y|\Lambda) \tag{9}$$

After computing predictions as probabilistic training labels using Equation 9, we determine cluster partitions from predictions or label vectors computed. All the data points having the same labels are supposed to be in the same cluster. For instance, consider label vector calculated as {1, 3, 2, 1, 3, 4, 4, 1, 2} for nine data points {A, B, C, D, E, F, G, H, I}. In this case, four clusters are created as {A, D, H}, {C, I}, {B, E}, and {F, G}. These clusters formed are passed as inputs for generating a summary. The approach used to extract summary for clusters is explained in detail in Section 4.3.

4.3 Summary Generation

After obtaining the final clustering, tweets are extracted to generate the final summary. A summary should contain most of the information without any redundancy. Hence, a set of tweets of varying characteristics should be selected, such that the summary contains most of the information. We have ranked tweets in each cluster. Then, clusters are ranked based on the average score of the tweets inside a cluster. Top-ranked tweets from each cluster are selected in an extractive way considering rank-wise clusters.

4.3.1 Tweet Ranking

For ranking of tweets, firstly, we have computed the tweet's score in each cluster using a weighted sum of two 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 kth cluster have M tweets, $\{t_1, t_2, \ldots, t_M\}$, then, tweet-scoring feature for a tweet t_l is described below

1) MaxSumTFIDF ($F1_{t_l}^k$): 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.
2) CountNamedEntities ($F2_{t_l}^k$): In disaster event, named entity recognition plays a major role [62] as it identifies location, organization, numerals, and many more. To implement it, we have used spaCy[26], an open-source python library designed for performing advanced Natural Language Processing (NLP) tasks. It includes a statistical entity recognition system that assigns labels to tokens in contiguous spans. Here, companies, places, organizations, and products are among the named, and numeric entities which are identified using the same model.

   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

$$F2_{t_l}^k = Count(NER_{t_l})/Q \tag{10}$$

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

Thus, the final score of tweet t_l in kth cluster will be

$$F_{t_l}^k = \frac{1}{2} \times F1_{t_l}^k + \frac{1}{2} \times F2_{t_l}^k \tag{11}$$

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.

[26] https://spacy.io/usage/linguistic-features#named-entities

5 Experimental Setup

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

5.1 Datasets

For the purpose of experimentation in our summarization task, we have used four disaster events, including natural and human-made disasters that have 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 datasets[27] are used by the paper [21]. As these datasets are designed for real-time tweet summarization; therefore, gold summaries are provided at two breakpoints of 2000 and 5000 tweets.

5.2 Experimental Settings

1) For DBSCAN clustering algorithm, min_samples was set to 5, and epsilon was selected from a pool of values {0.2, 0.3, 0.4, 0.5, 0.6, 0.7}. Value of epsilon is selected from the pool, so that silhouette score[28] is maximized.
2) For algorithms mentioned in Section 4.1 other than DBSCAN, they take the *number of clusters to be formed* as a parameter. We have set it to the number of clusters present in the clustering solution obtained by DBSCAN.
3) For classification tasks, the parameters used are described in Table 3.

To show the effect of the number of clusters in our framework, we have shown a case study in Section 7 by varying the number of clusters in the range of $[5, 50]$ with an interval of 5.

5.3 Comparative Methods

We have considered MOOTweetSumm [2] and COWTS [21] approaches for comparison purpose. MOOTweetSumm employs multi-objective optimization for microblog summarization where statistical measures refecting the quality of the summary, namely the TF-IDF score, anti-redundancy, length of the tweets, are optimized using the search capability of a multi-objective differential evolution technique. COWTS classifies tweets as situational or non-situational and then summarizes situation tweets. COWTS approach focuses on extracting tweets with the maximum number of content words (nouns, numerals, and verbs). In addition to COWTS, we have also presented several baselines of our proposed approach *GenSumm* as discussed below:

- LexRank[29]: LexRank is an unsupervised graph based commonly used approach for automatic text summarization where graph method is exploited to score sentences.
- Luhn Summarizer[30]: It is a naive summarization approach based on TF-IDF and sentences are ranked based on keyword frequency and proximity within a sentence.
- LSA Summarizer[31]: Latent Semantic Analysis is a relatively new algorithm which combines term frequency with singular value decomposition.
- TextRank[32]: 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.
- SumBasic[33]: It utilizes words that are frequently occurring in a document to generate a summary. This method is often used as a baseline in the literature.
- KL-Sum: In [63], authors have proposed a text summarization algorithm that finds a set of sentences whose overall length is less than L words and the unigram distribution is similar to the source document.
- Reduction[34]: It is a graph-based summarization, where a sentence salience is computed as the sum of the weights of its edges to other sentences.
- Bert Extractive Summarizer[35]: The algorithm first embeds the sentences, then executes a clustering algorithm. It finds the sentences that are closest to the cluster's centroids to generate a summary.
- Summa[36]: It also uses TextRank but with optimizations on similarity functions.

5.4 Evaluation Measures

We have used the well-known ROUGE-N metric to assess the summary quality, which counts the N-gram overlapping words between predicted and actual summary. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially a set of metrics for evaluating automatic summarization of texts as well as machine translations. We used ROUGE-L F-score proposed in [64], the Longest Common Subsequence (LCS) based statistics. Note that baseline papers have reported ROUGE-1 F-score that is the overlap of unigrams between the system generated summary and the reference summary as an evaluation measure. However, by using Rouge-1, only the count of the number of single word overlapping between the predicted summary and the ground truth is considered. This seems to

[27] http://cse.iitkgp.ac.in/ krudra/disaster_dataset.html.
[28] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html
[29] https://pypi.org/project/lexrank/
[30] https://en.wikipedia.org/wiki/Sentence_extraction
[31] http://www.kiv.zcu.cz/jstein/publikace/isim2004.pdf
[32] https://pypi.org/project/textrank/
[33] https://www.cs.bgu.ac.il/elhadad/nlp09/sumbasic.pdf
[34] https://iq.opengenus.org/graph-based-approach-for-text-summarization/
[35] https://pypi.org/project/bert-extractive-summarizer/
[36] https://pypi.org/project/summa/

TABLE 10
Comparison of Rouge-L F-Score Obtained by Our Summarization Approach and Existing Approaches Using Situational Tweet Streams at Breakpoints of 2000 and 5000 Tweets

	ROUGE-L F-score							
Datasets	HBlast		UkFlood		SHShoot		Hagupit	
Approach↓/Breakpoints →	**0-2000**	**2000-5000**	**0-2000**	**2000-5000**	**0-2000**	**2000-5000**	**0-2000**	**2000-5000**
LexRank	0.3854	0.3097	0.3273	0.2483	0.4000	0.3714	0.2102	0.1990
Luhn Summarizer	0.3174	0.2432	0.2634	0.2468	0.4816	0.3518	0.2330	0.1592
LSA Summarizer	0.4855	0.2419	0.2812	0.1711	0.4000	0.3551	0.1730	0.1953
TextRank	0.3618	0.2616	0.1940	0.2228	0.3798	0.3784	0.1765	0.1485
SumBasic	0.4762	0.3655	0.3125	0.2471	0.4621	0.4203	0.2804	0.2426
KL-Sum	0.3529	0.2833	0.3118	0.2579	0.2275	0.1837	0.2366	0.2260
Reduction	0.3379	0.2621	0.1883	0.2222	0.3906	0.3784	0.1917	0.1514
Bert Extractive Summarizer	0.4464	0.3518	0.2543	0.1969	0.4477	0.3948	0.2304	0.1663
Summa	0.2353	0.2527	0.2564	0.2284	0.3983	0.3539	0.1902	0.1275
COWTS	0.5610	0.3824	0.3412	0.2341	0.4645	0.2911	0.3013	0.3329
MOOTweetSumm	0.5543	0.3793	0.3513	0.2567	0.5000	0.4001	0.3513	0.3309
GenSumm	**0.5990**	**0.4493**	**0.3616**	**0.2614**	**0.5139**	**0.4013**	**0.4000**	**0.3399**

be a straightforward objective given all tweets are around a particular topic. Hence, we have utilized ROUGE-L F-score for evaluation purposes.

In ROUGE-L Score, the union of Longest Common Subsequence $LCS_\cap(ri, C)$ is calculated between reference summary sentence *ri* and every candidate summary sentence, *cj*. For example, consider *ri* = *w1 w2 w3 w4 w5 w6* and C contains two sentences: *c1* = *w1 w2 w6 w7 w8* and *c2* = *w1 w3 w8 w9 w5*. The longest common sub-sequence of *ri* and *c1* is *w1 w2 w6* and that of *ri* and *c2* is *w1 w3 w5*. The union is *w1 w2 w3 w5 w6* and therefore, $LCS_\cap(ri, C) = 5/6$. ROUGE-L F-Score is calculated as shown below:

$$R_{lcs} = \frac{\sum_{i=1}^{u} LCS_\cap(ri, C)}{m} \tag{12}$$

$$P_{lcs} = \frac{\sum_{i=1}^{u} LCS_\cap(ri, C)}{n} \tag{13}$$

$$F_{lcs} = \frac{(1+\beta^2) * R_{lcs} * P_{lcs}}{R_{lcs} + \beta^2 * P_{lcs}} \tag{14}$$

Here β is set to a very large number.

6 Experimental Results

In this section, we have described the results of our *GenSumm* approach on the datasets discussed in Section 5.1. Note that the authors of *COWTS* method have reported ROUGE-1 F-Score; therefore, we have executed the code of *COWTS* to obtain the results.

6.1 Summarization Results

In Table 10, a comparison of our proposed algorithm for tweet summarization is shown with respect to *COWTS* and other baselines as discussed in Section 5.3, at two breakpoints of 2000 and 5000 tweets. All algorithms are unsupervised in nature. It is evident that our approach performs better than existing ones. Considering mean Rouge-L F-score over all datasets at both breakpoints, our method improves by 5.16% and 2.53% over *COWTS* and *MOOTweetSumm*, respectively.

To illustrate the nature of summary, we have shown an example of generated summary in Table 12 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). This generated system has Rouge-L F-score of 0.5990.

6.2 Discussion of Results

This section has discussed the results obtained using the proposed *GenSumm* approach and the existing clustering algorithms. Figures showing ROUGE-L scores attained by various clustering approaches, given the same summarization framework, are presented in supplementary sheet, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TAFFC.2021.3131516. Individual clustering algorithm may perform better than the ensemble clustering solution. For instance, using agglomerative hierarchical clustering solution on HBlast dataset gives a ROUGE-L Score of 0.6469 for 2000 breakpoint and *GenSumm* attains a score of 0.5990. Though, for the Hagupit dataset, a ROUGE-L score of 0.2816 is obtained using agglomerative hierarchical clustering solution at 2000 breakpoint whereas *GenSumm* attains a score of 0.3999. Note that over time the quality of the clusters may decrease due to diversity among the tweets; therefore, we can consider recalculation of the

TABLE 11
Ranks of Various Clustering Algorithms Averaged Over Four Datasets at Both Breakpoints

Breakpoint ↓ Approach →	DBSCAN	K-Means	Agglomerative	Spectral	Birch	Gaussian Mixture	GenSumm
2000	3.250	4.500	4.250	6.250	3.500	4.250	1.750
5000	1.750	4.750	4.250	4.000	5.000	5.500	2.750
Average	2.500	4.625	4.250	5.125	4.250	4.875	2.250
FINAL RANK	**2**	**5**	**3**	**7**	**3**	**6**	**1**

TABLE 12
One of the Best System Summaries Generated by Our Proposed Approach for HBlast Dataset and Gold Summary at 5000 Breakpoint

Predicted Summary:
Help Numbers , Dhanalaxmi Ambulance Services at Dilshuknagar , +91 9391351543 , 9963857749 , 9440379926 . UPDATE , AFP , police say seven people have died and 47 people hurt in bomb blasts in Indian city of Hyderabad . 2 blasts reported near bus stand in southern Indian city of Hyderabad , 10 people feared dead , at least 40 others . State Helplines for Hyderabad Blasts , 040 27854771 , 040-27853408 040 27852435-36 . Hyderabad , Blast Needs A-ve blood , 3 units , For , Vishwanath At , Narayana Hrudayalaya , Suraram , Jeedimetla . BREAKING , Twin blast in Hyderabad's Dilsukh Nagar suburb , reports of 15 deaths , over 50 injured . Hyderabad serial blasts , at least 15 dead , 50 injured . Injured in Hyderabad Blasts being taken to Osmania General govt hospital , For info or to report any Afzalganj PS , 040-278 . Seven killed in Hyderabad blast , Seven people were killed and three others injured in an explosion in a crowd . Deadly blasts hit India's Hyderabad , Police say at least seven people killed and nearly 50 have been injured in three . Two deadly blasts in Dilsukh Nagar in Hyderabad , several casualties , 50 reported injured with 10 people died . At least 10 killed and dozens reportedly injured in two Hyderabad blasts . Alert in all major cities Mumbai, Kerala, Karnataka, Delhi across India . Breaking News , Twin blasts in Hyderabad , 7 people killed , 20 injured Casualties expected to rise . NIA , NSG teams flying to Hyderabad blast site . Hyderabad , Feb 21 , PTI , Nine people were killed and 32 injured tonight when at least three powerful serial blasts . List of hospitals in Dilsukhnagar , Hyderabad . 11 killed , 50 injured , says Home Minister Sushil Kumar Shinde . Explosion took place near venkatdri theatre in dilsukhnagar . Dilshuknagar Hospitals , Sigma 40-67120218 , Good Life 49640328 , Vasan eye 43400200 , HariPrasad 2404673 . Blood banks Dilsuknagar , SLMS 040-64579998 , Kamineni 39879999 , Hima Bindu 9246373536 , Balaji 2457219 .

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 bast 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 Dilsukhnagar , 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 Dilsukhnagar 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 .

cluster to avoid this. Hence it is not easy to choose a single clustering algorithm listed in Section 4.1, but their ensemble using generative modelling always outperforms state-of-the-art models. It happens because clustering performance by an algorithm is highly dependent on data distribution, and a single algorithm cannot be relied on for best performance. To check the superiority of the summarization results (Table 10) generated by our proposed approach (*GenSumm*), we have performed the statistical significance t-test similar to as discussed in Section 3.7. It has been noticed that all p-values fall below the 5% significance level which proves the statistical significance of our obtained results.

Further, we have also computed the ranks of different clustering solutions and the proposed approach for datasets mentioned in Section 5.1 at both the break-points. We have computed the average ranks as shown in Table 11. None of the existing clustering algorithms has ranked 1. Thus it is difficult to rely on a single clustering algorithm. Hence, ensembling of clusters is required as done in the proposed *GenSumm* approach.

6.3 Error Analysis

The performance of generative modelling is significantly dependent on the labeling functions used to tune the model. In the proposed GenSumm approach, we have used six labeling functions for each of the clustering algorithms: {DBSCAN, Aggomolerative Clustering, K-Means, Spectral Clustering, Gaussian Mixture Models, Birch Clustering}. As shown in Table 11, Spectral Clustering and Gaussian Mixture Models perform poorly. We have removed labeling functions for both Spectral Clustering (GenSummWithoutSpectral) and Gaussian Mixture one by one to explore the effect of labeling functions. The results for both the experiments are reported in Table 13. ROUGE-L F-Score varies with datasets and breakpoints without any pattern. Thus, changing the labeling function changes the partition of the dataset. Since the proposed summarization module's performance is significantly affected by the partitioning/clustering of the dataset, we have observed a notable change in the results. We have considered all six clustering approaches for labeling functions as they provide better results on ROUGE-L F-Score than the state-of-the-art COWTS method.

6.4 Time Analysis

It is important to get information fast during crisis. We have computed time taken to generate summary in various events and presented in Table 15. We have calculated time taken for each step (preprocessing, feature extraction, classification, and summarization) for 10 different events obtained from publicly available dataset, *HumAID: Human-Annotated Disaster Incidents Data from Twitter*[37] [65]. All the experiments were performed on CPU and we believe that computation time will decrease if GPU are utilized[66].

[37] https://crisisnlp.qcri.org/humaid_dataset

TABLE 13
Comparison of ROUGE-L F-Score on Summarization Task Using Various Approaches

Dataset →	HBlast		Hagupit		SHShoot		UKFlood	
Approaches ↓ Breakpoint →	2000	5000	2000	5000	2000	5000	2000	5000
GenSummWithoutGuassian	0.6294	0.4308	0.3320	0.3061	0.5155	0.4024	0.2795	0.2603
GenSummWithoutSpectral	0.5497	0.4133	0.2952	0.2998	0.5000	0.3925	0.2795	0.2787
GenSumm	0.5990	0.4493	0.4000	0.3399	0.5139	0.4013	0.3616	0.2614

TABLE 14
Comparison of Rouge-L F-Score Obtained by Varying Number of Clusters From 5 to 50 and DBSCAN Approach Using Situational Tweet Streams at Breakpoints of 2000 and 5000 Tweets of Four Datasets

#clusters ↓	ROUGE-L F-Score on Generative Modelling								Rank
Dataset →	HBlast 2000	HBlast 5000	Hagupit 2000	Hagupit 5000	SHShoot 2000	SHShoot 5000	UKFlood 2000	UKFlood 5000	
5	0.4485	0.3776	0.3360	0.2349	0.4231	0.3472	0.2884	0.2471	9
10	0.5000	0.3352	0.2141	0.2127	0.3946	0.3750	0.2398	0.2624	11
15	0.5118	0.2629	0.2578	0.2481	0.4286	0.3676	0.2895	0.1984	10
20	0.5222	0.3906	0.2620	0.2547	0.4944	0.4000	0.3224	0.1602	6
25	0.5360	0.4342	0.3038	0.3114	0.4324	0.3526	**0.3775**	0.2500	2
30	0.5567	**0.4696**	0.2574	0.2863	0.4920	0.3862	0.2779	0.2306	3
35	**0.6266**	0.4473	0.3094	0.2578	0.4267	0.2911	0.2431	0.2661	5
40	0.5420	0.4304	0.2381	0.2561	0.4401	0.3500	0.2792	0.2980	6
45	0.4754	0.3874	0.3070	0.2757	0.4368	**0.4305**	0.2787	0.2510	4
50	0.5583	0.3652	0.2719	0.2756	0.4986	0.3761	0.2529	0.2156	6
$\mathbb{C}$	0.5990	0.4493	**0.4000**	**0.3399**	**0.5139**	0.4013	0.3616	**0.2614**	**1**

Here, RANK refers to ranking of approach averaged over four datasets on two breakpoints; $\mathbb{C}$ *denotes that number of clusters is determined using DBSCAN.*

TABLE 15
Time Taken for Summary Generation by Proposed GenSumm Approach

		Time taken in seconds				
Event Name ↓	#Tweets	Preprocessing	Feature Generation	Classification	Summarization	Total
2017 Hurricane Irma	9399	14.10	438.81	0.74	1525.52	1979.18
2016 Kaikoura Earthquake	2195	4.80	103.72	0.14	99.62	208.28
2017 Sri Lanka Floods	560	1.90	26.33	0.09	22.47	50.79
2019 Cyclone Idai	3933	10.80	280.31	0.32	533.87	825.29
2017 Hurricane Maria	7278	11.80	347.96	0.37	765.47	1125.60
2016 Ecuador Earthquake	1563	3.20	68.02	0.11	56.25	127.58
2016 Hurricane Matthew	1654	3.50	79.28	0.12	64.18	147.08
2017 Mexico Earthquake	2015	4.00	90.96	0.15	97.16	192.27
2017 Hurricane Harvey	9112	13.60	427.65	0.39	1374.86	1816.50
2016 Italy Earthquake	1201	2.70	52.32	0.11	35.35	90.47

Preprocessing includes time taken for fragmentation. Feature generation refers to the time taken for generating sentence embeddings from BERTweet model.

7 Case Study on DBSCAN Algorithm

In our proposed *GenSumm* framework, DBSCAN clustering algorithm is used to determine the number of clusters. Hence, it might affect the parameters for other clustering approaches. To investigate this, we have executed the proposed approach for a fixed number of clusters (varying between 5 to 50 with an interval of 5) on all summarization datasets at two breakpoints. The aim is that all the clustering solutions should have same number of clusters. We can't control the number of clusters in DBSCAN and hence removed it. Also, reogranization of membership matrix returns one-to-one mapping between cluster labels of two different partitions. Clustering labels returned by DBSCAN can't be reorganized if the number of clusters is not same as that of reference clustering label.

The results obtained are shown in Table 14. From this table, it is evident that determining the number of clusters using DBSCAN algorithm has outperformed other experiments in most of the cases. Having fixed number of clusters might not fit on every dataset. Thus, DBSCAN is able to determine the number of clusters according to the dataset.

8 Conclusion and Future Works

The current article presents a simple, robust and a novel framework for classification followed by summarization to handle the tweet streams posted during disaster events that can help disaster management authorities provide immediate relief during havoc. The efficiency of the BERT model to solve various NLP problems has been proved in the literature. The proposed approach utilizes features from BERTweet for classification and summarization purposes. Multi-task learning is employed to simultaneously solve information classification and sentiment analysis for tweets that has improved the model's learning capability. In summarization, generative modelling is utilized as a smart ensemble to combine various weak supervised clustering approaches. Improved performance in terms of ROUGE-L F-Score of the proposed method compared to state-of-the-art systems proves the proposed approach's efficiency. In the future, we would like to build an unsupervised approach to classify tweets as situational or non-situational. Further, we would like to work on methods to ensemble various summarization approaches.

References

[1] N. Saini, S. Saha, and P. Bhattacharyya, "Microblog summarization using self-adaptive multi-objective binary differential evolution," *Appl. Intell.*, vol. 52, pp. 1686–1702, 2022.

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

[3] K. Rudra, A. Sharma, N. Ganguly, and M. Imran, "Classifying and summarizing information from microblogs during epidemics," *Inf. Syst. Front.*, vol. 20, no. 5, pp. 933–948, Oct. 2018.

[4] N. Saini, S. Saha, S. Mansoori, and P. Bhattacharyya, "Automatic parameter selection of granual self-organizing map for microblog summarization," in *Proc. Int. Conf. Neural Inf. Process.*, 2020, pp. 680–692.

[5] D. Q. Nguyen, T. Vu, and A. T. Nguyen, "BERTweet: A pre-trained language model for English Tweets," in *Proc. Conf. Empir. Methods Natural Lang. Process. Syst. Demonstrations*, 2020, pp. 9–14.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv: 1810.04805*.

[7] M.-T. Nguyen, A. Kitamoto, and T.-T. Nguyen, "Tsum4act: A framework for retrieving and summarizing actionable tweets during a disaster for reaction," in *Proc. Adv. Knowl. Discov. Data Mining*, 2015, pp. 64–75.

[8] N. Saini, S. Saha, S. Mansoori, and P. Bhattacharyya, "Fusion of self-organizing map and granular self-organizing map for microblog summarization," *Softw. Comput.*, vol. 24, no. 24, pp. 18 699–18 711, Dec. 2020.

[9] D. Miller, "Leveraging BERT for extractive text summarization on lectures," 2019, *arXiv:1906.04165*.

[10] S. Bano, B. Divyanjali, A. Virajitha, and M. Tejaswi, "Document summarization using clustering and text analysis," *Int. J. Eng. Technol.*, vol. 7, 2018, Art. no. 456.

[11] K. Shetty and J. S. Kallimani, "Automatic extractive text summarization using k-means clustering," in *Proc. Int. Conf. Elect., Electron., Commun., Comput., Optim. Techn.*, 2017, pp. 1–9.

[12] D. Wang, S. Zhu, T. Li, Y. Chi, and Y. Gong, "Integrating document clustering and multidocument summarization," *ACM Trans. Knowl. Discov. Data*, vol. 5, no. 3, pp. 1–26, Aug. 2011.

[13] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1866–1881, Dec. 2005.

[14] J. Al-Shaqsi and W. Wang, "A clustering ensemble method for clustering mixed data," in *Proc. Int. Joint Conf. Neural Netw.*, 2010, pp. 1–8.

[15] X. Wu, T. Ma, J. Cao, Y. Tian, and A. Alabdulkarim, "A comparative study of clustering ensemble algorithms," *Comput. Elect. Eng.*, vol. 68, pp. 603–615, 2018.

[16] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Lang. Process.*, 2009, pp. 1003–1011. [Online]. Available: https://www.aclweb.org/anthology/P09–1113

[17] A. Agrawal, A. An, and M. Papagelis, "Learning emotion-enriched word representations," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 950–961. [Online]. Available: https://www.aclweb.org/anthology/C18–1081

[18] 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*, 2014, pp. 159–162.

[19] S. Verma, "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.

[20] I. Varga et al., "Aid is out there: Looking for help from tweets during a large scale disaster," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, 2013, pp. 1619–1629.

[21] 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.*, 2015, pp. 583–592.

[22] 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.

[23] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*.

[24] J. Longhini, C. Rossi, C. Casetti, and F. Angaramo, "A language-agnostic approach to exact informative tweets during emergency situations," in *Proc. IEEE Int. Conf. Big Data*, 2017, pp. 3739–3475.

[25] S. Madichetty and M. Sridevi, "Detecting informative tweets during disaster using deep neural networks," in *Proc. 11th Int. Conf. Commun. Syst. Netw.*, 2019, pp. 709–713.

[26] R. ALRashdi and S. O'Keefe, "Deep learning and word embeddings for tweet classification for crisis response," 2019, *arXiv: 1903.11024*.

[27] 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. Modelling Decis. Mak.*, 2016, pp. 255–268.

[28] 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, 2019.

[29] S. S. Haykin, S. S. Haykin, S. S. Haykin, and S. S. Haykin, *Neural Networks and Learning Machines*. Upper Saddle River, NJ, USA: Pearson, 2009.

[30] S. S. Ray, A. Ganivada, and S. K. Pal, "A granular self-organizing map for clustering and gene selection in microarray data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1890–1906, Sep. 2016.

[31] C. De Maio, G. Fenza, V. Loia, and M. Parente, "Time aware knowledge extraction for microblog summarization on twitter," *Inf. Fusion*, vol. 28, pp. 60–74, 2016.

[32] 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.

[33] 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, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230817302279

[34] C. De Maio, G. Fenza, V. Loia, and M. Parente, "Time aware knowledge extraction for microblog summarization on twitter," *Inf. Fusion*, vol. 28, pp. 60–74, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S156625351500055X

[35] 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.*, 2015, pp. 583–592.

[36] 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.

[37] M. Dragoni, C. da Costa Pereira, A. G. Tettamanzi, and S. Villata, "Smack: An argumentation framework for opinion mining," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 4242–4243.

[38] M. Dragoni, "A three-phase approach for exploiting opinion mining in computational advertising," *IEEE Intell. Syst.*, vol. 32, no. 3, pp. 21–27, May/Jun. 2017.

[39] F. Alam, H. Sajjad, M. Imran, and F. Ofli, "Standardizing and benchmarking crisis-related social media datasets for humanitarian information processing," 2020.

[40] A. Olteanu, S. Vieweg, and C. Castillo, "What to expect when the unexpected happens: Social media communications across crises," in *Proc. 18th ACM Conf. Comput. Supported Cooperative Work Social Comput.*, 2015, pp. 994–1009.

[41] A. Olteanu, C. Castillo, F. Diaz, and S. Vieweg, "Crisislex: A lexicon for collecting and filtering microblogged communications in crises," in *Proc. 8th Int. AAAI Conf. Web Social Media*, 2014, pp. 376–385.

[42] M. Imran, P. Mitra, and C. Castillo, "Twitter as a lifeline: Human-annotated twitter corpora for NLP of crisis-related messages," in *Proc. 10th Int. Conf. Lang. Res. Eval.*, 2016, pp. 1638–1643.

[43] M. Imran, S. M. Elbassuoni, C. Castillo, F. Diaz, and P. Meier, “Extracting information nuggets from disaster-related messages in social media,” in *Proc. 12th Inf. Syst. Crisis Response Manage.*, 2013, pp. 791–801.

[44] F. Ofli, F. Alam, and M. Imran, “Analysis of social media data using multimodal deep learning for disaster response,” 2020, *arXiv:2004.11838*.

[45] F. Alam, F. Ofli, and M. Imran, “CrisisMMD: Multimodal twitter datasets from natural disasters,” *Proc. Int. AAAI Conf. Web Social Media*, vol. 12, no. 1, 2018.

[46] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.

[47] C. Hutto and E. Gilbert, “VADER: A parsimonious rule-based model for sentiment analysis of social media text,” in *Proc. Int. AAAI Conf. Web Social Media*, 2015, pp. 216–225.

[48] S. A. Qureshi, G. Dias, M. Hasanuzzaman, and S. Saha, “Improving depression level estimation by concurrently learning emotion intensity,” *IEEE Comput. Intell. Mag.*, vol. 15, no. 3, pp. 47–59, Aug. 2020.

[49] A. Dobrescu, M. V. Giuffrida, and S. A. Tsaftaris, “Doing more with less: A multitask deep learning approach in plant phenotyping,” *Front. Plant Sci.*, vol. 11, 2020, Art. no. 141. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpls.2020.00141

[50] M. Shad Akhtar, D. Ghosal, A. Ekbal, P. Bhattacharyya, and S. Kurohashi, “A multi-task ensemble framework for emotion, sentiment and intensity prediction,” 2018, *arXiv: 1808.01216*.

[51] S. A. Qureshi, S. Saha, M. Hasanuzzaman, and G. Dias, “Multitask representation learning for multimodal estimation of depression level,” *IEEE Intell. Syst.*, vol. 34, no. 5, pp. 45–52, Sep./Oct. 2019.

[52] K. Thung and C.-Y. Wee, “A brief review on multi-task learning,” *Multimedia Tools Appl.*, vol. 77, pp. 29705–29725, 2018.

[53] A. F. Agarap, “Deep learning using rectified linear units (ReLU),” 2018.

[54] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jun. 2014.

[55] D. Nguyen, K. Ali Al Mannai, S. Joty, H. Sajjad, M. Imran, and P. Mitra, “Robust classification of crisis-related data on social networks using convolutional neural networks,” in *Proc. Int. AAAI Conf. Web Social Media*, 2017, pp. 632–635. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14950

[56] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” in *Proc. Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2016, pp. 427–431.

[57] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19–1423

[58] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter,” 2019, *arXiv:1910.01108*.

[59] Y. Liu et al., “RoBERTa: A robustly optimized bert pretraining approach,” 2019, *arXiv:1907.11692*.

[60] A. Ratner, S. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, “Snorkel: Rapid training data creation with weak supervision,” *VLDB J.*, vol. 29, 2020, Art. no. 269.

[61] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, “Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 991–1005, Oct. 2009.

[62] 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. Retrieval*, 2018, pp. 265–274.

[63] A. Haghighi and L. Vanderwende, “Exploring content models for multi-document summarization,” in *Proc. Hum. Lang. Technol. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2009, pp. 362–370. [Online]. Available: https://www.aclweb.org/anthology/N09–1041

[64] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Proc. Workshop Text Summarization Branches Out*, 2004, pp. 74–81. [Online]. Available: https://www.aclweb.org/anthology/W04–1013

[65] F. Alam, U. Qazi, M. Imran, and F. Ofli, “Humaid: Human-annotated disaster incidents data from twitter with deep learning benchmarks,” in *Proc. Int. AAAI Conf. Web Social Media*, vol. 15, pp. 933–942, 2021.

[66] E. Buber and B. Diri, “Performance analysis and CPU versus GPU comparison for deep learning,” in *Proc. 6th Int. Conf. Control Eng. Inf. Technol.*, 2018, pp. 1–6.

Diksha Bansal is currently working toward the BTech degree in computer science and engineering with the Indian Institute of Technology Patna. Her research interest lies in deep learning, natural language processing and explainable AI.

Rahul Grover is currently working toward the BTech degree in computer science and engineering with the Indian Institute of Technology Patna. His research interest lies in deep learning, natural language processing and explainable AI.

Naveen Saini (Member, IEEE) received his PhD degree from the Department of Computer Science and Engineering, Indian Institute of Technology, Patna, India. He is currently an assistant professor in the Department of Information Technology at Indian Institute of Information Technology Allahabad. He has also worked as an Assistant Professor in the Department of Computer Science at IIIT Lucknow. Before joining IIIT Lucknow, he has worked as a researcher at 4IR Applied Research Center and Assistant Professor at Endicott College of International Studies, Woosong University, South Korea. He was a postdoctoral fellow with IRIT (Institut De Recherche En Informatique De Toulouse) which is a joint research unit of 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. More details about him can be found at https://sites.google.com/view/nsaini.

Sriparna Saha (Senior Member, IEEE) received the masters' and PhD degrees in computer science from Indian Statistical Institute Kolkata, India, in the years 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 is the recipient of various prestigious awards like Google India Women in Engineering Award, 2008, NASI Young Scientist Platinum Jubilee Award 2016, BIRD Award 2016, IEI Young Engineer's Award 2016, etc. For more information, refer to https://www.iitp.ac.in/sriparna/.