Multi-Task Learning approach to identify sentences with impact and affected location in a disaster news report

Sumanta Banerjee[1,4], Shyamapada Mukherjee[2], Sivaji Bandyopadhyay[3]

[1]National Institute of Technology Silchar, India,

[2]National Institute of Technology Rourkela, India,

[3]Jadavpur University, Kolkata, India,

[4]Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India

Correspondence: sumanta.banerjee85@gmail.com

Abstract

The first priority of action in the Sendai Framework for Disaster Risk Reduction 2015-2030 advocates the understanding of disaster risk by collecting and processing practical information related to disasters. A smart collection may be the compilation of relevant and summarized news articles focused on some key pieces of information such as disaster event type, geographic location(s), and impacts. In this article, a Multi-Task Learning (MTL) based end-to-end model has been developed to perform three related tasks: sentence classification depending on the presence of (1) relevant locations and (2) impact information to generate a summary, and (3) identification of the causes or event types in disaster news. Each of the three tasks is formulated as a multilabel binary classification problem. The results of the proposed MTL model have been compared with three popular transformer models: BERT, RoBERTa, and ALBERT. It is observed that the proposed model showed better performance scores than the other models in most cases.

1 Introduction

The first priority of action of the third United Nations (UN) World Conference on Disaster Risk Reduction (WCDRR)[1] advocates disaster risk understanding through the collection and processing of relevant and practical pieces of information. News reports published by reputed sources provide fast and reliable information that can be processed and used to keep track of such events Rossi et al., 2018; Chen and Wang, 2022.

Researchers found that Caruana, 1996; Sun et al., 2020 jointly learning multiple related tasks (Multi-task learning) benefits the learning of each of them. The knowledge gathered through the training of one task is used in learning others. It helps the model improve its generalization ability for all related tasks and reduces model overfitting on training data Almeida and Martins, 2013; Thung and Wee, 2018.

Experiments have shown that small language models can summarize well Ghinassi et al., 2024 if used on a specific category. Note that, the proposed language model[2] is light (around 29 million parameters and 115 MB size), simple, and has been designed for a precise category of documents. This work aims to do three tasks: classification of sentences depending on (1) disaster location and (2) impact information (Table 2), and (3) classify a document on nine themes (event present/absent, covid, flood, storm, heavy rain, cloudburst, landslide, earthquake, tsunami - as shown in Table 1). The union of the sentences extracted from the above two pieces of information is considered the summary of the disaster news article. The above three tasks are learned by homogeneous feature MTL Zhang and Yang, 2021 based encoder-decoder model that takes an array of the words in a document as input and learns all three tasks simultaneously by sharing the word encoder layer output among them. The sentence extraction tasks are performed by a decoder architecture that is attentive towards the important sentence features, and a fully connected decoder performs the multi-label document classification task. The design has four main components: a word encoder followed by three decoders that share the encoder outputs. The encoder encodes an array of tokens/words in a document. The encoded words are passed to the event class decoder, which classifies the document into nine classes, i.e., themes. The encoded words of each sentence are turned into sentence encoding and passed to two identical attention-based decoders that classify the sentences in the document based on location and impact information. The Bahdanau attention Bahdanau et al., 2014 mechanism (instead

[1]https://www.undrr.org/media/16176/download

[2]https://github.com/RanaBan/DL-Experiments/blob/master/event_location_impact.ipynb

Proceedings of the Fourth Workshop on NLP for Positive Impact (NLP4PI), pages 229–238
July 31, 2025

of the self-attention Vaswani, 2017) has been used here, which suits the design and the small (7692 documents containing 126125 sentences and 45085 unique tokens) dataset [3] (described in Banerjee et al., 2023a). Besides the proposed MTL model, the performances of the component classifiers are separately tested to do the ablation study. The method has shown impressive results (Ref. section 6) on each task.

The rest of the paper is organized as follows: a literature review is presented in *section 2*. The methodology is covered in *section 3*.The training and inference of the proposed model are discussed in *section 4* and *section 5*, respectively. A discussion of the results and analysis of the outputs is given in *section 6*. Finally, the article is concluded in *section 7*.

2 Related work

The proposed model is designed to generate a disaster news extractive summary with location and impact sentences following the Multi-task Learning (MTL) approach. The methods in Banerjee et al., 2023b; Nafi et al., 2020 includes the disaster impacts and causes in the generated abstractive summary. The NER (Named Entity Recognition) (Imran et al., 2013; Lingad et al., 2013; Fernandes et al., 2021), machine learning Téllez Valero et al., 2009 and statistical techniques Panem et al., 2014 are applied to extract the disaster impacts from tweet and news texts. There are excellent works that extract salient information from text (not limited to disaster-related reports). The MTL based abstractive summarization methods in Kirstein et al., 2022; Xu et al., 2020; Lu et al., 2019; Isonuma et al., 2017 and Chen et al., 2019 jointly learn the target summarization task with other language understanding tasks. Interestingly, the extractive methods in Jia et al., 2020 applied the graph attention network (GAT) and in Qiu et al., 2020 used automatic classification based on geoscience-dictionary attention. The MTL model in Mulyar et al., 2021 learns eight tasks on clinical notes and Huang et al., 2022 learns four tasks across multiple language datasets. There are MTL models identifying event information Lv et al., 2022, summarizing legal documents Agarwal et al., 2022, efficiently generating sentence embeddings Lamsiyah et al., 2023, and processing conversation Song et al., 2023. The authors in Aguirre and Dredze, 2024 dealt with the performance disparity in models on different data subpopulations by transferring demographic fairness transfer among related tasks.

The literature shows that the MTL-based approach is highly efficient when employed in closely related tasks. The methods targeting summarization have used various language understanding task(s) as auxiliary. However, an extractive summarization method that learns multiple sentence classification tasks on related topics (impacts and relevant location) is rarely present in the literature. The proposed model does the above and also the relevant event identification task together in an end-to-end model. An NER technique may find "flood" disaster in *"...complaints flood T.N. police. . . "*. However, the event identifier is intended to find none in it.

3 Methodology

The end-to-end model depicted in Figure 1 starts with a token embedding layer followed by a layer encoding the sequence of tokens. The encoded sequence is then sent to the event class decoder for event identification. The encoded token sequence in each sentence is averaged and sent to the attention-based decoders that classify each sentence based on the (impact and relevant location) information it carries. Recurrent neural network (RNN) is highly efficient when processing sequential data. The Long Short-Term Memory (LSTM) neural network Hochreiter and Schmidhuber, 1997 is a category of RNNs that efficiently addresses the exploding and vanishing gradient issue of RNN training. In this method, the encoder processes a sequence of sentences in a document and the decoder uses the contextual information from the encoder and produce a sequence of labels. Therefore, the LSTM units are employed to construct the encoder and decoder structure of the model.

3.1 Embedding layer

The input to the proposed model is an array of $M \times N$ token indices $(t_1, t_2, ..., t_{(M\times N)})$. The embedding layer converts each $t_i (1 \leq i \leq (M \times N))$ to a suitable vector representation of embedding dimension (EmbDim = 128). The embedding function produces token embeddings $X(x_1, x_2, ..., x_{(M\times N)})$ and can be expressed as the following,

$$\text{Embedding: } t_i \in \mathbb{N}^1 \rightarrow x_i \in \mathbb{R}^{EmbDim} \text{ for each sample } T(t_1, t_2, ..., t_{(M\times N)})$$

[3] https://dx.doi.org/10.21227/hsdv-2t76

Sl No.	Document	Event	COVID	Flood	Storm	Heavy rain	Cloudburst	Landslide	Earthquake	Tsunami
1	Two more deaths 56 new COVID 19 cases in Gujarat. Two more persons died of coronavirus in Gujarat taking the death toll in the State to 30 the State Health Department said on Wednesday ...	1	1	0	0	0	0	0	0	0
2	Heavy rain leaves many roads water logged. Heavy rain was reported in several parts of the city and some places in the district on Sunday evening. The rain that started around 5 p.m. lashed the city for more than two hours ...	1	0	0	0	1	0	0	0	0

Table 1: Two labeled-documents on eight event classes

Sl No.	Sentence	Location	Impact
1	Heavy rain in Dakshina Kannada three electrocuted power supply hit.	1	1
2	Power supply severely affected MESCOM suffers Rs.	0	1
3	10 crore loss As rain and gusty winds continued unabated three persons were electrocuted in two incidents in Puttur taluk on Monday.	1	1
4	In the first incident Chandra and Kaushik died at Anchinadka in Kumbra section of MESCOM while they were carrying a wooden log from the forest Puttur Rural Police said.	1	1

Table 2: Labels of four sample sentences on location and impact information

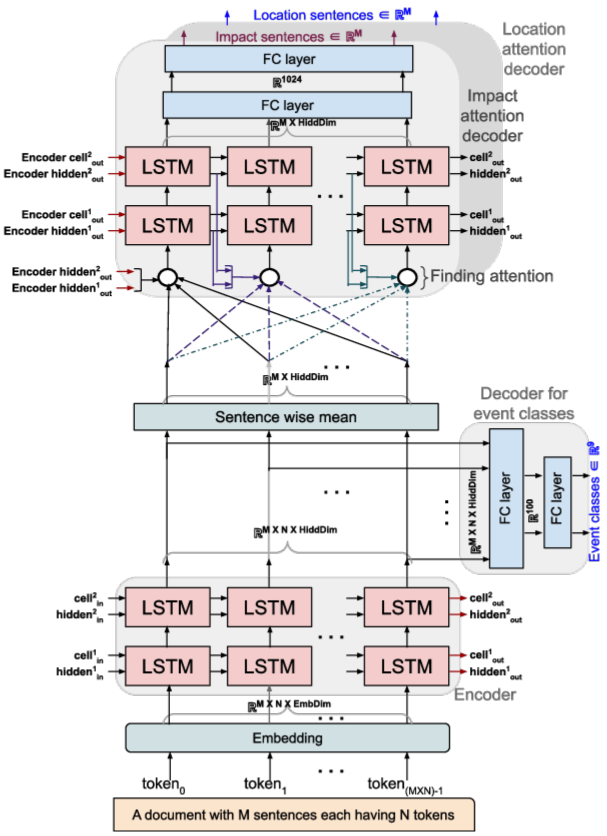

Figure 1: Block diagram of the proposed Multi-Task Learning model. All the components: the embedding layer, the encoder layer, the decoder for event classes, the layer for sentence representations (sentence wise mean), and the decoders for location and impact sentence classification, are explained in section 3

3.2 Encoder layer

The encoder is designed with a unidirectional 2-layer RNN-LSTM unit. The encoder generates an output of (HiddDimm in Figure 1) hidden-dimension ($H_D = 128$) in each time-step for each token embedding $x_i, 1 \leq i \leq (M \times N)$ (equation 1).

$$u_i \leftarrow EncoderLSTM_{time\ step=i}(x_i, u_{i-1}, c_{i-1}) \quad (1)$$

where $c \in \mathbb{R}^{H_D}$, $u \in \mathbb{R}^{H_D}$.
So, the final output $U = (u_1, u_2, ..., u_{(M \times N)})$ of the encoder after all the time steps has dimension $M \times N \times H_D$. The hidden-state and cell-state outputs of the last time step are also recorded.

3.3 Decoder for event classes

The decoder for the event classes is a fully connected two-layer neural network that takes the encoded sequence of dimension $M \times N \times H_D$ and produces a nine-class output for the nine binary labels. The first label signifies the presence or absence of any event in the sample document with 1 or 0, respectively. The rest of the eight labels indicate whether the sample has (1) "COVID-19", (2) "Storm", (3) "Flood", (4) "Heavy rain", (5) "Cloudburst", (6) "Landslide", (7) "Earthquake", and (8) "Tsunami" with 1 and 0. It is expressed using equation 2 where the first fully connected layer with the ReLU (rectified linear unit) activation function converts U to a ($a \in \mathbb{R}^d$, d=100). Then another fully connected layer with the sigmoid activation function converts a to E as a 9-dimensional vector of real numbers. The eight disaster classes are chosen after studying the corpus.

$$a \leftarrow \sigma(U^{1\times(M\times N\times H_D)} \times W_1^{(M\times N\times H_D)\times d} + b_1^{1\times d}),$$
$$E \leftarrow \sigma(a^{1\times d} \times W_2^{d\times 9} + b_2^{1\times 9}) \quad (2)$$

where $a \in \mathbb{R}^d$, and $E \in \mathbb{R}^9$. W_1, W_2, b_1, and b_2 are the weights and biases of the two layers.

3.4 Sentence representations from encoded sequence

The attention-based decoder finds relative importance among the sentences of a sample document. Hence, it requires sentence representations instead of tokens. In order to get the required sentence representations from the encoded sequence $U = (u_1, u_2, ..., u_{(M\times N)})$, each of the N consecutive encoded sequences that belongs to a sentence in the sample document are averaged (equation 3).

$$v_j \leftarrow mean(each\ N\ consecutive\ u\ vectors) \quad (3)$$

where $v_j \in \mathbb{R}^{H_D}$, and V(v_1, v_2, ..., v_M) whose each element represents a sentence. Then, the result is used in the attention-based decoders to classify sentences. The averaging is done in the following simple way. Let, $u_i = [x_1, x_2, ..., x_{H_D}]$ then v_j is calculated with equation 4.

$$v_j \leftarrow \left[\frac{1}{N}\sum_{i=1}^{N} x_{(i,1)}, \frac{1}{N}\sum_{i=1}^{N} x_{(i,2)}, \ldots, \frac{1}{N}\sum_{i=1}^{N} x_{(i,H_D)}\right] \quad (4)$$

3.5 Decoders for location and impact sentence classification

The *attention* mechanism used here is introduced by Bahdanau et al. Bahdanau et al., 2014 in Neural Machine Translation (NMT) model. The proposed method implements a similar attention technique that finds a set of relevant sentences in a document. The attention weights determine the relative importance of each of the sentences over other sentences. In this way, the model is guided to pay more attention to the relatively more important sentences for the tasks. An attention weight for each of the sentences is determined to make the decoder focus on the relevant position in the input document. A fully connected network is used as the alignment model that takes the decoder hidden state from the previous time step concatenated with the sentence representation v_i, $1 \leq i \leq M$ to find the importance score $attention_i$ of v_i. The generated scores for each of the sentences $attention \in \mathbb{R}^M$ is then passed through the *softmax* function. Now, each $attention_i$ value of the result $attention$ vector is multiplied by its corresponding sentence representation vector v_i. It suppresses some parts and also boosts other parts of the sentence representations $V(v_1, v_2, \ldots, v_M)$ that are unimportant and important, respectively, for the output on the t^{th} time step. Finally, a fully connected network is used to find the $atten\_applied$ input for the t^{th} time step of RNN-LSTM from the product of $attention$ and $V(v_1, v_2, \ldots, v_M)$. Both attention-based decoders for the location and impact sentence identification tasks follow the same procedure delineated in the Algorithm 1.

4 Training

All the samples are shuffled to properly mix the contents. Then they are divided into the train, validation, and test sets (8:1:1) with 6153, 769, and 770 samples. The distribution of the event type labels has been shown in Table 3. The losses of all three tasks are calculated using the Binary Cross Entropy (BCE) function. The equation 5 presents the BCE function that finds the loss from the predicted label $\hat{Y}(\hat{y}_1, \hat{y}_2, ..., \hat{y}_N)$ and true label $Y(y_1, y_2, ..., y_N)$.

$$BCE\_loss \leftarrow \frac{-1}{N}\sum_{i=1}^{N}(y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)) \quad (5)$$

Two different procedures are used to calculate the loss, one for the event identification task and another for the sentence labeling tasks. The loss (l_1) of the event identification task is calculated in the following steps.

1. Let y and $\hat{y}$ are the true and predicted binary labels for events for a sample
2. If the event present/absent bit $Y[0]$ is 0 then, $loss \leftarrow BCE\_loss(\hat{y}_0, y_0)$ [If the document actually has no event then only the first predicted bit is compared with the first ground truth bit.]
3. else, $loss \leftarrow BCE\_loss(\hat{Y}, Y)$ [Otherwise, all predicted bits are compared with all the ground truth bits.]

On the other hand, the losses of the sentence labeling tasks (l_2, and l_3) are separately calculated by equation 6.

$$loss \leftarrow BCE\_loss(\hat{Y}[0\ to\ SC], Y[0\ to\ SC]) \quad (6)$$

Algorithm 1 Algorithm for impact and location sentence decoders

Require: V, $EncHidd$, $EncCell$
Ensure: $sentence\_labels$

```
hidd ← mean(EncHidd_1, EncHidd_2)        ▷ Last time step hidden outputs from 2-layer encoder
for each i in M do                                                        ▷ M = length(V)
    for each v_j in V do
        weights[j] ← FC_attn_weights(concat(hidd, v_j))                   ▷ weights[j] ∈ ℝ^1
    end for
    attention ← softmax(weights)
    for each a_j and v_j in attention and V do
        attn_applied[j] ← a_j × v_j                                   ▷ a_j-scalar and v_j-vector
    end for
    attn_input ← relu(FC_apply_attn(attention_applied))               ▷ attn_input ∈ ℝ^{H_D}
    if i is 0 then                                                    ▷ For the first time step
        output, DecHidd, DecCell ← rnnLSTM(attn_input, EncHidd, EncCell)
    else                                                              ▷ For other time steps
        output, DecHidd, DecCell ← rnnLSTM(attn_input, DecHidd, DecCell)
    end if
    hidd ← mean(DecHidd_1, DecHidd_2)
    DecOutputs[i] ← output
end for
intermediate ← relu(FC_intermediate(DecOutputs))                  ▷ intermediate ∈ ℝ^1024
sentence_labels ← σ(FC_sent_class(intermediate))                 ▷ sentence_labels ∈ ℝ^M
```

	covid	flood	storm	heavy rain	cloudburst	landslide	earthquake	tsunami
Train	2127 (31.34%)	2130 (31.39%)	633 (9.33%)	1211 (17.85%)	380 (5.6%)	36 (0.53%)	162 (2.39%)	107 (1.58%)
Validate	282 (32.79%)	263 (30.58%)	72 (8.37%)	165 (19.19%)	42 (4.88%)	7 (0.81%)	16 (1.86%)	13 (1.51%)
Test	271 (32.0%)	276 (32.59%)	93 (10.98%)	144 (17.0%)	34 (4.01%)	6 (0.71%)	13 (1.53%)	10 (1.18%)

Table 3: The total number of times each label has appeared (maximum once for a document), and its share in each section of the dataset is given (a document may have multiple events).

Finally, the loss quantities from the document classification and two sentence classifications are averaged to get the loss of the MTL model ($MTL\ model\ loss = \frac{l_1+l_2+l_3}{3}$, where event, impact related sentence and location related sentence identification losses are l_1, l_2, and l_3, respectively). At the time of data preparation, after going through the sentence and token frequencies, the sentences per document and tokens per sentence are fixed at M (40) and N (20). There are shorter documents having fewer sentences than M. The variable SC represents $sentence\ count$ (equation 6) that carries the number of sentences for shorter documents and M for bigger documents. So, the loss calculation in equation 6 is done only on the actual length of the sentence, and it helps to avoid calculating loss for the padded sentence entries. Due to its relevance to classification tasks, the F-measure scores are considered for judging the best architecture among the models with 1-layer, 2-layer, and 3-layer LSTM encoder and decoder units. Each of the above three generated low precision and high recall values, which means a high false-positive ratio. On the basis of the results found in Table 4, the architecture with the 2-layer LSTM is selected as it resulted in the least false positive ratio. As the 3-layered architecture resulted worse than the 2-layered one, it is assumed that a further increase in the number of LSTM layers in the encoder and decoder units would not improve the results. The MTL model and its component classifiers are separately trained with $batchsize = 20$, $epochs = 5$, $dropout = 0.5$, *adamw* optimizer function Loshchilov and Hutter,

	1-layer			2-layer			3-layer		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
event	0.684	0.653	0.668	**0.805**	**0.767**	**0.775**	0.619	0.546	0.58
location	0.123	0.618	0.205	**0.261**	**0.865**	**0.401**	0.157	0.982	0.271
impact	0.084	0.46	0.142	**0.328**	**0.736**	**0.454**	0.106	0.895	0.19

Table 4: The impact of 1-layer, 2-layer, and 3-layer LSTM encoder and decoder units on the model performance in terms of F-measure (highest scores are in boldface).

2017 with $weight\ decay$ 0.01 (L2 regularizer) for the best results.

5 Inference

The trained MTL model and the component classifiers are separately applied to the test dataset with 770 samples. At first, the tokens in a sample are converted into embeddings (Ref. section 3.1). Then the sequence encoder generates token encodings from the embeddings (Ref. section 3.2). The encoded sequence is then passed through the event class decoder to identify the probable events in the sample document (Ref. section 3.3). The encoded sequence of tokens is then converted to an encoded sequence of sentences (Ref. section 3.4). After that, the encoded sequence of sentences is used in both the attention decoders (Ref. section 3.5). Finally, the predicted labels for the event classes and the sentence classifications are used to map the actual event names and the actual sentences in the test set. The results found after comparing the predicted and ground truth labels are elaborated in section 6.

6 Results and Discussion

This section shows results obtained after using the custom disaster news dataset to train and test the proposed MTL model, each of the component classifiers (*for the ablation study*) in the proposed MTL model, and three pretrained popular transformer models: Bidirectional Encoder Representations from Transformers (BERT) Devlin et al., 2019, A Lite BERT (ALBERT) Lan et al., 2019, and Robustly Optimized BERT Pretraining Approach (RoBERTa) Liu et al., 2019. All the above three transformers are pre-trained on BookCorpus and English Wikipedia datasets, whilst RoBERTa is further trained on CommonCrawl-News, OpenWebText, and Stories datasets. All these pre-trained transformer models can be fine-tuned for downstream language understanding tasks. In this work, the pre-trained bert-base-uncased, roberta-base, and albert-base-v2 models (from the huggingface library Wolf et al., 2020), each with 12 layers, 12 heads, and 768 hidden dimensions (L=12, A=12, and H=768), have been fine-tuned and tested on the event identification and sentence classification tasks.

The dice-coefficient (equation 7) finds the overlap or similarity between each pair of values in the 0 to 1 range between two equal-length vectors Guindon and Zhang, 2017. A score close to ‘1’ indicates high similarity in them. Table 5 shows the dice-coefficient scores of the MTL model and the component models.

$$dice\ coefficient(\vec{y}, \vec{\hat{y}}) = \frac{2 \times \sum \parallel y_i \cdot \hat{y_i} \parallel}{\sum y_i + \sum \hat{y_i}} \quad (7)$$

The event class label of a sample consists of nine bits. The first bit signifies the presence/absence of an event. It is used in the loss calculation of event identification task (Ref. section 4) Table 7. Depending on the context of the document it may also identify a crisis event that is not present in the list of events (“fire”, or “lightning”). The sentence identification accuracy (Table 6) is the mean of the ratio of correct prediction and total sentences in each document. After that, the mean of all the documents is taken as the average accuracy. The average accuracy of identifying each of the event classes is calculated by the mean of the number of correct predictions with respect to the total number of samples (Table 8). The precision, recall and F1 scores are calculated for each instance (samples average) using scikit learn library Pedregosa et al., 2011 and their averages are shown in Table 9 for all the models (where TP, TN, FP, and FN stand for True Positive, True Negative, False Positive, and False Negative, respectively).

The proposed MTL model showed impressive results in identifying whether an event is present in a sample document (Table 7). The mean dice-coefficients presented in Table 5 show a good result by the proposed MTL model in identifying the presence of the eight different disaster types. Between the other two tasks, the proposed MTL model performed well in impact sentence identification. The predicted real values are rounded off and converted

	proposed-MTL	ablation study	BERT	RoBERTa	ALBERT
event classification	**0.715**	0.668	0.493	0.498	0.478
impact sentences	**0.591**	0.548	0.369	0.359	0.359
location sentences	**0.526**	0.442	0.335	0.332	0.317

Table 5: Mean dice coefficient of ground truths and predictions (highest scores are in boldface)

	proposed-MTL	ablation study	BERT	RoBERTa	ALBERT
impact sentences	**0.627**	0.602	0.509	0.505	0.512
location sentences	**0.675**	0.653	0.531	0.527	0.530

Table 6: Mean accuracy of sentence extraction (highest scores are in boldface)

	proposed-MTL	ablation study	BERT	RoBERTa	ALBERT
Accuracy	**0.869**	0.791	0.806	0.811	0.769

Table 7: Mean accuracy of event present/absent identification (highest score is in boldface)

	proposed-MTL	ablation study	BERT	RoBERTa	ALBERT
COVID	**0.882**	0.810	0.544	0.544	0.549
Storm	**0.925**	0.913	0.539	0.540	0.536
Flood	**0.893**	0.848	0.808	0.812	0.839
Heavy rain	**0.896**	0.825	0.648	0.649	0.674
Cloudburst	0.992	0.981	**0.994**	0.993	**0.994**
Landslide	**0.953**	0.907	0.861	0.867	0.878
Earthquake	**0.992**	0.962	0.941	0.944	0.951
Tsunami	**0.987**	0.954	0.973	0.976	0.970

Table 8: Mean accuracy of event type identification (highest scores are in boldface)

	proposed-MTL			ablation study			BERT			RoBERTa			ALBERT		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Eve ident	0.805	0.767	**0.775**	0.741	0.757	0.749	0.482	0.506	0.494	0.491	0.514	0.502	0.486	0.473	0.479
Imp sent	0.328	0.736	**0.454**	0.251	0.720	0.372	0.397	0.497	0.441	0.370	0.455	0.408	0.371	0.451	0.407
Loc sent	0.261	0.865	0.401	0.162	0.836	0.271	0.400	0.431	**0.415**	0.387	0.428	0.406	0.356	0.386	0.370

Table 9: Precision, Recall and F1 scores of the proposed-MTL and component classifiers (highest scores are in boldface)

to binary values to calculate accuracy. Tables 6 and 8 show the average accuracies. Calculated scores in the first and second tables are all above 0.6 and 0.8, which is good. The proposed MTL model has shown a good precision score in event class prediction and moderate scores in the other two. The model has also shown good recall for all three tasks. The BERT model for location sentence labeling has shown a little better F1 score than the proposed MTL model. All transformers have shown balanced precision and recall scores. Overall, in most of the scores, the proposed MTL model has shown the best performance among all the experimented models.

7 Conclusion

This article introduces an MTL based model that jointly learns identifying (1) disaster event types and the sentences containing the (2) disaster locations and (3) impacts in a disaster news article. The union of the set of extracted sentences forms a summary. Eight frequent disaster events are identified from the corpus and used as the target labels. Three component classifiers of the proposed MTL model and three transformer models are tested on the same data to compare the performances. The MTL model has performed well in comparison to the component models and transformer models (Ref. section 6). Hopefully, the model can perform better if it is trained with a larger amount of samples. A relevant dataset in multilingual news would be prepared, and the generalization ability of the proposed model on this dataset would be tested in future.

7.1 Limitation

In this section, two sample outputs are discussed to demonstrate the limitations of the proposed MTL

1	**News text**	Heavy rain likely on Saturday too. Many parts of the State continued to receive heavy rainfall on Friday even as the northeast monsoon remained active. Among the areas that received the heavy rain were Meenambakkam and Avinasi 9 cm each Perundurai Nungambakkam and Kalpakkam 8 cm each Poonamallee and Ponneri 7 cm each. Educational institutions in Chennai and the neighbouring districts of Kancheepuram and Tiruvallur remained closed . An official of the Meteorological Department said an upper air cyclonic circulation over Sri Lanka and the adjoining Gulf of Mannar and other areas persisted. Many parts of the State both in the north and the south would receive heavy rain on Saturday too. Chief Minister O. Panneerselvam held a meeting with senior Ministers and officials at the Secretariat. Chief Secretary Mohan Verghese Chunkath and Commissioner of Revenue Administration T.S. Sridhar briefed him on the situation. Secretaries of various departments who were appointed monitoring officers for the districts were asked to visit the rain hit areas...
	Event class	'Heavy rain'
	Location sentences	'Many parts of the State continued to receive heavy rainfall on Friday even as the northeast monsoon remained active', 'Among the areas that received the heavy rain were Meenambakkam and Avinasi 9 cm each Perundurai Nungambakkam and Kalpakkam 8'
	Impact sentences	'Many parts of the State continued to receive heavy rainfall on Friday even as the northeast monsoon remained active', 'Educational institutions in Chennai and the neighbouring districts of Kancheepuram and Tiruvallur remained closed', 'Many parts of the State both in the north and the south would receive heavy rain on Saturday too', 'Secretaries of various departments who were appointed monitoring officers for the districts were asked to visit the rain hit areas', 'While some of them have left Chennai for their respective districts others are on their way an official said', 'The monitoring officers were also advised to oversee the preparations of the district administration', 'People could contact the State Emergency Operations Centre the control room in districts Tamil Nadu Generation and Distribution Corporation and'
2	**News text**	Lack of money for stormwater drain network ups flood risk. Officials unable to convince funding agencies for the past few years Delay in getting funds for creating a network of stormwater drains and canals in Kosasthalaiyar basin and Kovalam basin continues to be a challenge to monsoon preparedness in most parts of the city. The Kosasthalaiyar basin comprises areas such as Tiruvottiyur Manali and Madhavaram while the Kovalam basin consists of neighbourhoods along the East Coast Road and Rajiv Gandhi Salai. Even as Chennai Corporation officials have claimed that the city is prepared for the monsoon no improvement in stormwater drain or canal network has been made in the past five years after the proposal for the ₹3000 crore project was made in 2012. Areas such as Tiruvottiyur Manali Madhavaram Perungudi and Sholinganallur are likely to face floods like the one in 2015. Ex councillors of some wards in northern and southern parts of the city said the delay in getting funding for the project was affecting many neighbourhoods in Chennai. DPR in final stage The detailed project report for stormwater drains in Kosasthalaiyar basin is in the final stages of preparation. We are exploring funding options. We have proposed to the Asian Development Bank and the Japan International Cooperation Agency said a senior Corporation official...
	Event class	'–'
	Location sentences	'Ex councillors of some wards in northern and southern parts of the city said the delay in getting funding for'
	Impact sentences	'Officials unable to convince funding agencies for the past few years Delay in getting funds for creating a network of', 'Even as Chennai Corporation officials have claimed that the city is prepared for the monsoon no improvement in stormwater drain'

Table 10: Two examples demonstrate the performance and limitations of the model

model. The example outputs are shown in Table 10. In the first example, the model has captured two sentences as location sentences. Between them, the first one may be selected for the token "State" which the model may have wrongly identified as a location. The model missed the other sentences containing "Sri Lanka", "Chennai", and "Tamilnadu". It may be the reason that those sentences have less event-related information and do not have that relative importance or attention. The impact sentence identification task has captured the sentences having "heavy rainfall" related information. However, it selected some sentences that contain information about how government officials are monitoring the situation and what people can do in an emergency situation, which may not be considered an impact related information. The second news article is about getting funds to build a stormwater drainage system. The event identifier has found no events in it, which may be a good prediction. However, the sentence selected as a location sentence has no location information in it, which may be selected due to the token "city". It missed the sentences "The Kosasthalaiyar basin comprises areas such as Tiruvottiyur Manali…", and "Even as Chennai Corporation officials have…" with locations like "Tiruvottiyur", "Manali", and "Chennai" mentioned in them. It may be the reason that the location sentence identifier could not get event-related information in those sentences. The impact sentence identifier selected two sentences that talked about the delay in getting funds for the drainage system and the claims made by officials. This prediction should have been empty. Noticeably, the sentences having both impact and location information have a higher chance of selection. The model confuses the words that come with a location, like city or state, with a real location. In some documents, there is no impact/location sentence, but the model selects some as relevant. Hopefully, an increased amount of training data would improve its performance.

References

Abhishek Agarwal, Shanshan Xu, and Matthias Grabmair. 2022. Extractive summarization of legal decisions using multi-task learning and maximal marginal relevance. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1857–1872. Association for Computational Linguistics.

Carlos Alejandro Aguirre and Mark Dredze. 2024.

Transferring fairness using multi-task learning with limited demographic information. In *Proceedings of the Third Workshop on NLP for Positive Impact*, pages 32–49. Association for Computational Linguistics.

Miguel B Almeida and Andre FT Martins. 2013. Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 196–206.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Sumanta Banerjee, Shyamapada Mukherjee, and Sivaji Bandyopadhyay. 2023a. Disaster-news datasets for multi-label document classification, sentence classification, and abstractive document summarization tasks. In *2023 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, pages 1–6.

Sumanta Banerjee, Shyamapada Mukherjee, Sivaji Bandyopadhyay, and Partha Pakray. 2023b. An extract-then-abstract based method to generate disaster-news headlines using a dnn extractor followed by a transformer abstractor. *Information Processing and Management*, 60(3):103291.

Rich Caruana. 1996. Algorithms and applications for multitask learning. In *ICML*, pages 87–95. Citeseer.

Chun-Chieh Chen and Hei-Chia Wang. 2022. Using community information for natural disaster alerts. *Journal of Information Science*, 48(5):718–732.

Yangbin Chen, Yun Ma, Xudong Mao, and Qing Li. 2019. Multi-task learning for abstractive and extractive summarization. *Data Science and Engineering*, 4:14–23.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Chelsea Fernandes, Joshua Fernandes, Sharon Mathew, Shubham Raorane, and Anuradha Srinivasaraghavan. 2021. Automated disaster news collection classification and geoparsing. In *Proceedings of the International Conference on Smart Data Intelligence (ICSMDI 2021)*, volume 5, pages 42–46.

Iacopo Ghinassi, Leonardo Catalano, and Tommaso Colella. 2024. Efficient aspect-based summarization of climate change reports with small language models. In *Proceedings of the Third Workshop on NLP for Positive Impact*, pages 123–139. Association for Computational Linguistics.

Bert Guindon and Ying Zhang. 2017. Application of the dice coefficient to accuracy assessment of object-based image classification. *Canadian Journal of Remote Sensing*, 43(1):48–61.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zhiqi Huang, Milind Rao, Anirudh Raju, Zhe Zhang, Bach Bui, and Chul Lee. 2022. Mtl-slt: multi-task learning for spoken language tasks. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 120–130.

Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. 2013. Practical extraction of disaster-relevant information from social media. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*, pages 1021–1024. International World Wide Web Conferences Steering Committee / ACM.

Masaru Isonuma, Toru Fujino, Junichiro Mori, Yutaka Matsuo, and Ichiro Sakata. 2017. Extractive summarization using multi-task learning with document classification. In *Proceedings of the 2017 Conference on empirical methods in natural language processing*, pages 2101–2110.

Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631. Association for Computational Linguistics.

Frederic Thomas Kirstein, Jan Philip Wahle, Terry Ruas, and Bela Gipp. 2022. Analyzing multi-task learning for abstractive text summarization. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 54–77. Association for Computational Linguistics.

Salima Lamsiyah, Abdelkader El Mahdaouy, Saïd El Alaoui Ouatik, and Bernard Espinasse. 2023. Unsupervised extractive multi-document summarization method based on transfer learning from bert multi-task fine-tuning. *Journal of Information Science*, 49(1):164–182.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

John Lingad, Sarvnaz Karimi, and Jie Yin. 2013. Location extraction from disaster-related microblogs. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013,* *Companion Volume*, pages 1017–1020. International World Wide Web Conferences Steering Committee / ACM.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yao Lu, Linqing Liu, Zhile Jiang, Min Yang, and Randy Goebel. 2019. A multi-task learning framework for abstractive text summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9987–9988.

Jianwei Lv, Zequn Zhang, Li Jin, Shuchao Li, Xiaoyu Li, Guangluan Xu, and Xian Sun. 2022. Trigger is non-central: Jointly event extraction via label-aware representations with multi-task learning. *Knowledge-Based Systems*, 252:109480.

Andriy Mulyar, Ozlem Uzuner, and Bridget McInnes. 2021. MT-clinical BERT: scaling clinical information extraction with multitask learning. *Journal of the American Medical Informatics Association*, 28(10):2108–2115.

Nasik Muhammad Nafi, Avishek Bose, Sarthak Khanal, Doina Caragea, and William H. Hsu. 2020. Abstractive text summarization of disaster-related documents. In *17th International Conference on Information Systems for Crisis Response and Management, ISCRAM 2020, May 2020*, pages 881–892. ISCRAM Digital Library.

Sandeep Panem, Manish Gupta, and Vasudeva Varma. 2014. Structured information extraction from natural disaster events on twitter. In *Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval & Reasoning, Web-KR@CIKM 2014, Shanghai, China, November 3, 2014*, pages 1–8. ACM.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Qinjun Qiu, Zhong Xie, Liang Wu, and Liufeng Tao. 2020. Dictionary-based automated information extraction from geological documents using a deep learning algorithm. *Earth and Space Science*, 7(3):1–18.

C. Rossi, F.S. Acerbo, K. Ylinen, I. Juga, P. Nurmi, A. Bosca, F. Tarasconi, M. Cristoforetti, and A. Alikadic. 2018. Early detection and information extraction for weather-induced floods using social media streams. *International Journal of Disaster Risk Reduction*, 30:145–157.

Kaisong Song, Yangyang Kang, Jiawei Liu, Xurui Li, Changlong Sun, and Xiaozhong Liu. 2023. A speaker turn-aware multi-task adversarial network for joint user satisfaction estimation and sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13582–13590.

Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. 2020. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems*, 33:8728–8740.

Alberto Téllez Valero, Manuel Montes y Gómez, and Luis Villaseñor Pineda. 2009. Using machine learning for extracting information from natural disaster news reports. *Computación y sistemas*, 13(1):33–44.

Kim-Han Thung and Chong-Yaw Wee. 2018. A brief review on multi-task learning. *Multimedia Tools and Applications*, 77(22):29705–29725.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Weiran Xu, Chenliang Li, Minghao Lee, and Chi Zhang. 2020. Multi-task learning for abstractive text summarization with key information guide network. *EURASIP Journal on Advances in Signal Processing*, 2020:1–11.

Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609.