Urgency Detection in Social Media Texts Using Natural Language Processing

Navya Makkena[1], Dr.ABM Rezbaul Islam[2], Dr.Cihan Varol[3], Dr.Min Kyung An[4]
Department of Computer Science
Sam Houston State University, Huntsville, TX 77340, USA
nxm074@shsu.edu[1], ari014@shsu.edu[2], cxv007@shsu.edu[3], an@shsu.edu[4]

2024 IEEE 18th International Conference on Semantic Computing (ICSC) | 979-8-3503-8535-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICSC59802.2024.00031

***Abstract*— Social media platforms are used extensively to request emergency help during large-scale natural or humanitarian disasters. Machine learning can be used best to mobilize resources in any disaster. Natural language processing comes to our rescue to help detect urgency, identifying certain expressions, words, and phrases that typically showcase concern or immediate need. A deep learning model can be implemented, which can be used to classify the messages as a particular urgent class label. First responders or governments can use this urgency class label to categorize the requests to provide relief measures to needy people. We have used the hugging face disaster messages dataset and reclassified messages based on urgency. We leveraged the advantages of both CNN and LSTM to get high-level features and long-term dependencies. We developed a model combining Convolutional neural networks (CNN) and bidirectional long short-term memory(BiLSTM) networks. We also conducted experiments with various model architectures, such as ensemble and hierarchical, along with varying word embeddings. We observed the performance of fine-tuning pre-trained models such as BERT and DistilBERT with our dataset. We evaluated that the developed model using CNN and BiLSTM performs better with only 10% of trainable parameters comparable to pre-trained models such as BERT, which has 110M trainable parameters. We could also observe that general-purpose word embeddings such as GloVe give comparable results to domain-specific word embeddings.**

***Keywords*— Convolutional neural networks, Deep learning, Emergency services, Hazards, Long short-term memory**

I. Introduction

Fires, flooding, landslides, epidemics, road accidents, earthquakes, violent winds, crises, and disasters are part of our daily lives. They define how we interact amongst ourselves and with the natural environment. Social media is also continuously redefining how we interact with ourselves in these modern times. Access and coverage are unlike at any point in human history. Social media is used during various disasters to disseminate relevant information to a large audience to report their well-being, ask for help, or give updates about the ongoing situation. We can leverage social media ubiquity and the power of Machine learning to improve our lives and provide timely aid to those in crisis. In rapidly changing situations worldwide, urgency detection can help health workers, first responders, and policymakers predict and respond quickly to emergencies. Disasters can happen anytime, anywhere, whether a global pandemic, terror incident, or natural disaster. With the ability to detect urgency automatically from a live social media stream, a system could lead first responders and law enforcement to a volatile situation even before an emergency call. Detecting urgency can help people who need medical help and save lives.

The urgency detection model can filter the tweets or messages from social media to prioritize the most urgent and forward them to different types of first responders to address those during crises and save the lives of people in danger. Implementing natural language processing models capable of distinguishing requests for help on Twitter would thus aid in the emergency response to large-scale disasters by allowing the public or governments to identify better and assist those who need urgent help. Urgency detection has been successfully used in cases such as law enforcement, humanitarian crises, and health care hotlines to "flag up" text that indicates a certain urgency threshold.

Various studies have been conducted to classify tweets using various machine-learning methods. Some studies have explored traditional text features used in natural language processing (NLP), including n-grams and lexical resources [1]. Others have worked towards a deep convolutional neural network that exploits character- to sentence-level information to perform sentiment analysis of short texts [2] and a recurrent convolutional neural network for text classification without human-designed features[3]. Most of the Machine-learning classification tasks conducted on disaster or crisis-related tweets have addressed whether tweets are about the crises or not [4], the type of disaster [4], the type of information being conveyed [4,5], informativeness [6], and general sentiment polarity[7]. Using the machine learning classification tasks mentioned above, first responders can cherry-pick useful tweets from a large dataset, but none of them tackle the problem of explicitly detecting urgent tweets that should immediately need a first response (i.e., calls for help or reports of specific people requiring help from first responders). The closest labeling scheme for urgency or help needed was done between urgent and non-urgent tweets by Imran et al. [5], in which classifications nearer to urgent or non-urgent are made, i.e., the "Injured or dead people" and "Missing, trapped, or found people" categories include tweets of people requesting help or reporting that specific people need help. However, even these categories must be narrower to be directly helpful to first responders. Urgent tweet classification done by Devaraj et al.

979-8-3503-8535-9/24/$31.00 ©2024 IEEE
DOI 10.1109/ICSC59802.2024.00031

[9] is the closest method to classifying tweets as urgent or non-urgent. However, this research mainly focused on a dataset related to 2017's Hurricane Harvey and is also a binary classification task. In this research, we want to build a machine-learning classification model for classifying disaster or crisis-related datasets containing more than two labels. Instead of classifying it as urgent or not, we want to add more information to the labels so that these classification labels will give information to first responders for prioritizing the tweets/messages. Using a subset of nearly 40000 messages from our dataset, we are building a model using Convolutional and Recurrent neural networks. The convolutional neural network extracts the features by applying relevant filters, and the Recurrent Neural Network analyzes these features, considering the information received from previous time steps. We want to evaluate the classifier's performance using precision, recall, and F1 score metrics.

II. Background & related work

Extensive research has been conducted on crisis informatics, disaster-related tweet classification, tweet sentiment analysis, and automatic detection of crisis messages. Firoj et al. [8] have consolidated eight human-annotated crisis-related datasets and provided 166.1k and 141.5k tweets for informativeness and humanitarian classification tasks, respectively. Twitter is the most used social media platform during crises. Imran et al. [5] studied helpful information on Twitter that can be used for situational awareness and other humanitarian disaster response efforts. They presented human-annotated Twitter corpora collected during 19 different crises that took place between 2013 and 2015. They trained machine learning classifiers to demonstrate the annotations' utility and published the first largest word2vec word embeddings trained on 52 million crisis-related tweets.

While Kouloumpis et al. [1] investigated the utility of linguistic features for detecting the sentiment of Twitter messages, they evaluated the usefulness of existing lexical resources and features that capture information about the informal and creative language used in microblogging. A deep convolutional neural network has also been studied to perform sentiment analysis of messages that exploit from character- to sentence-level information [2]. S. Lai et al. [3] studied recurrent convolutional neural networks for text classification [15] without human-designed features. Their model applied a recurrent structure to capture contextual information as far as possible when learning word representations, introducing considerably less noise than traditional window-based neural networks. While Wang et al. [10] studied convolutional recurrent neural network for text classification, which enjoys both the advantages of convolutional neural networks for extracting local features from text and those of recurrent neural networks (LSTM) in memory to connect the extracted features, Aytuğ Onan et al. [11] did research on a bidirectional convolutional recurrent neural network architecture, which utilizes two separate bidirectional LSTM and GRU layers, to derive both past and future contexts by connecting two hidden layers of opposite directions to the same context.

Many researchers have focused on detecting crisis-related messages (tweets)[12] on Twitter. They have started by showing the varying definitions of importance and relevance relating to disasters, leading to the concept of use case-dependent actionability that has recently become more popular [12]. Burel et al. [4] introduced Dual-CNN, a semantically enhanced deep learning model to target the problem of event detection in crises from social media data. They had added a layer of semantics to a traditional Convolutional Neural Network (CNN) model to capture the contextual information that is generally scarce in short, ill-formed social media messages. Kejriwal et al. [13] explored crisis-related short message research areas in the direction of urgency detection. They presented a robust, low-supervision social media urgency system that adapts to arbitrary crises by leveraging labeled and unlabeled data in an ensemble setting. Supervised(non-neural models) and deep learning models have been studied by Devaraj et al. [9] to create an urgency detection classifier using a subset of the dataset of 2017's Hurricane Harvey. They highlighted the utility of average word embeddings for training non-neural models and that such features produce competitive results with more traditional n-gram and POS features. While most of the research has been focused on English language-related tweets or short crisis messages, Efsun Sarioglu Kayi et al. [14] have researched low-resource languages using transfer learning approaches. They adopted cross-lingual embeddings constructed using different methods to extract features of the tweets, including a few state-of-the-art contextual embeddings such as BERT, RoBERTa, and XLM-R. They explored semi-supervised approaches using unlabeled tweets and experimented with an ensemble setting.

III. Methodology

Most of the crisis-related messages or tweets are in the form of sequences. We want to classify the social media tweets/messages into multiple class labels. Most research regarding text classification, Twitter sentiment analysis, and classifying tweets into urgent or not has been done using non-neural networks or Convolutional neural networks (CNNs). CNN is a worthy approach to extracting higher-level features invariant to local translation. Many proposals [10] were based on LSTM architecture in sequence-to-sequence problems such as Text Classification. Recurrent neural networks (RNNs), especially LSTM and Bidirectional LSTM networks, have been applied to natural language processing systems with comparative, remarkable results.

A. Problem Assertion

We are provided with a text from social media platforms, and we need to classify the message into one of the urgency class labels. The first responders will use this urgency informative label to cater to emergency needs. Let us consider the input sequence of text with M words and generate an output label, one of the target urgency class labels.

B. Data Collection

We have analyzed datasets for many crises and from sources such as CrisisNLP [5], CrisisLex, CrisisMMD [33], Disaster Response Data (DRD), and Disasters on social media (DSM). CrisisLex is one of the most extensive publicly available datasets, which consists of two subsets, i.e., CrisisLexT26[31] and CrisisLexT6[32]. CrisisLexT26

comprises data from 26 crisis events in 2012 and 2013 with annotations for informative vs. not informative and humanitarian categories(eight classes) classification tasks, among others. CrisisLexT6, on the other hand, contains data from six crisis events between October 2012 and July 2013 with annotations for related vs. not-related binary classification tasks [8]. CrisisNLP is another large-scale dataset collected during 19 different disaster events between 2013 and 2015 and annotated according to different schemes, including classes from humanitarian disaster response and some classes related to health emergencies [5]. After analyzing datasets for many different crises and from sources such as CrisisNLP, CrisisLex, CrisisMMD, Disaster Response Data (DRD), and Disasters on social media (DSM), all these datasets have been classified as informative vs. not informative and humanitarian categories such as Caution and advice, Displaced people, and evacuations among others. The problem we found in the CrisisNLP dataset is that, as the dataset is classified into different humanitarian categories, these categories contain some urgency-related tweets/messages, and some messages indeed provide information about the incidents, which may lead to misclassification. It will be cumbersome to classify the tweets/messages in the CrisisNLP dataset manually. We have a dataset in Hugging Face [16] which contains nearly 30,000 messages captured from different events such as an earthquake in Chile in 2010, super-storm Sandy in the USA in 2012, floods in Pakistan in 2010, and news articles of different disasters, etc., The dataset has been labeled with 36 different categories related to disaster response. As 36 different categories are too many class labels, we have attempted classifying the messages based on 36 different categories into 5 class labels based on urgency, i.e. (Not Related, Related to disaster but not related urgency, Related to Urgent AID needed(Basic survival goods immediately required), Related to Urgent AID needed(Medical Help or Basic things for survival), Related to Urgent AID needed(Regarding security or military or rescue operation, Missing people, Other).

C. Class Imbalance Problem

One potential problem we observed in the labeled dataset is class imbalance because most of the messages belong to non-urgency class labels, which means potential urgent messages are significantly fewer; this may lead to the model classifying every tweet as a majority class label. We have gone through different options to overcome the class imbalance problem, such as Random Over Sampling, SMOTE, ADASYN, Borderline SMOTE, and Text augmentation techniques. Creating synthetic message samples using Text augmentation techniques works better for this project. Data augmentation in the computer vision research area for images has got much research work. However, when it comes to text generation, it is still a grey area because it is hard to augment text due to the high complexity of language. We learned about TextGenie [17], a text data augmentation library that augments text datasets. This library uses various NLP methods such as paraphrasing using T5, BERT mask filling, and converting sentences to active or passive voice. When we tried augmenting the minority classes in the training dataset, the augmented data samples that we got were similar to the dataset in inputted training data samples, which may lead to overfitting the model because of duplicate samples. Later, we analyzed the nlpaug library [18] for augmenting text datasets. It consisted of character Augmenters, Word Augmenters, and sequential Augmenters. Character Augmenters augment data at a character level, while word augmenters do at a word level. Word augmenters work better for our project area as we want to augment the dataset by replacing words. These word augmenters use different word embeddings to find the most similar group of words to replace in the input training dataset. Contextual embeddings such as BERT use language models to predict the possible target words based on the sentence context. Out of many different word augmenters, we used ContextualWordEmbsAug, which feeds possible surrounding words to BERT, DistilBERT, RoBERTa or XLNet language model to find out the most suitable word for augmentation, SynonymAug, which substitute similar word according to WordNet/ PPDB synonym, BackTranslationAug which leverages two translation models for augmentation. We applied these augmenters to minority classes to resolve the class imbalance problem.

D. Dataset Pre-Processing

The pre-processing steps, along with data, are briefly mentioned in Fig. 1. For pre-processing the dataset, we have used Python libraries such as nltk, re, NumPy, pandas, bs4, sklearn, contractions, etc.,

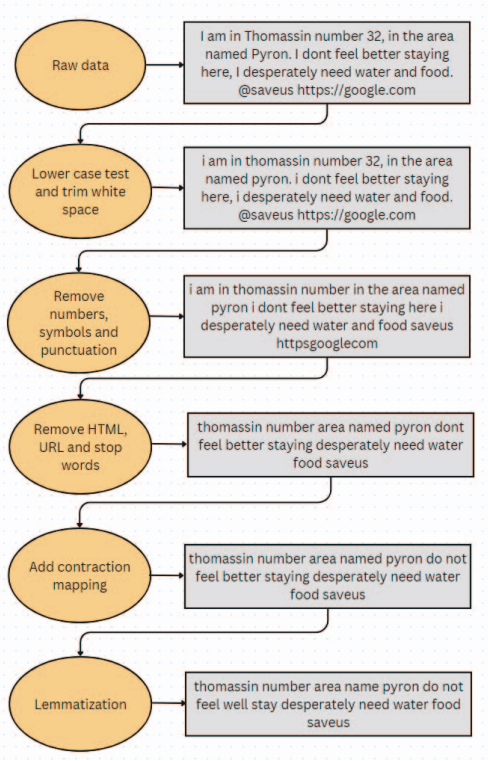

Fig. 1. Pre-processing steps

E. Deep Learning Architectures

a) Convolutional Neural Networks

Convolution Neural Networks(CNNs) are multi-layered artificial neural networks that can detect complex features in data, for instance, extracting features in image and text data. CNNs have been used in computer vision tasks such as image classification, object detection, and image segmentation.

However, recently, CNNs have been applied to text problems. Convolutional Neural Networks comprise two main layers, i.e., a convolution layer for obtaining features from the data and a pooling layer for reducing the feature map size. The output from these layers is then passed to fully connected layers to get the predictions. CNN's main work here is classifying a sentence into pre-determined categories by considering n-grams, i.e., its words, sequence of words, or characters or sequence of characters. 1-D Convolutions over text can be described below [25][29]. Given a sequence of words w_1, w_2, …, w_n, each is associated with an embedding vector of dimension d. A 1D convolution of width-k is the result of moving a sliding window of size k over the sentence and applying the same convolution filter or kernel to each window in the sequence, i.e., a dot-product between the concatenation of the embedding vectors in each window and a weight vector u, which is then often followed by a non-linear activation function g. Considering a window of words, the concatenated vector of the i^{th} window is then:

$$x_i = [\, w_i, w_{i+1}, \ldots, w_{i+k}] \;\in\; R^{k*d} \tag{1}$$

The convolution filter is applied to each window, resulting in scalar values r_i, each for the i^{th} window:

$$r_i = g(x_i * u) \in R \tag{2}$$

In practice, one typically applies more filters, u_1,….,u_m, which can then be represented as a vector multiplied by a matrix U and with an addition of a bias term b:

$$r_i = g(x_i * U + b) \tag{3}$$

With $r_i \in R^m$, $x_i \in R^{k*d}$, $U \in R^{k.d*m}$ *and* $b \in R^m$

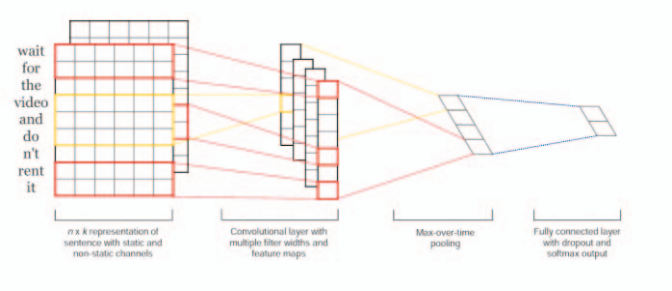

Fig. 2. The architecture of a sample Convolutional Neural network model for text classification. courtesy of Yoon Kim [25]

b) Recurrent Neural Networks

Recurrent neural networks (RNNs) are a special type of artificial neural networks adapted to work for time series data or data that involve sequences. Ordinary feedforward neural networks are only meant for data points that are independent of each other. However, suppose we have data in a sequence such that one data point depends upon the previous data point. In that case, we must modify the neural network to incorporate the dependencies between these data points. RNNs have the concept of "memory" that helps them store the states or information of previous inputs to generate the subsequent output of the sequence. However, RNNs suffer the vanishing gradient problem when they try to capture long-term dependencies.

c) Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. LSTMs are explicitly designed to avoid long-term dependency problems. At a high level, LSTM works very much like an RNN cell. There are three parts in an LSTM cell known as gates. The first part is called the Forget gate, the second is known as the Input gate, and the last is the Output gate. Just like a simple RNN, an LSTM also has a hidden state where h(t-1) represents the hidden state of the previous timestamp and h(t) is the hidden state of the current timestamp. In addition to that, LSTM also has a cell state represented by c(t-1) and c(t) for previous and current timestamps, respectively. Here, the hidden state is known as short-term memory, and the cell state is known as long-term memory.

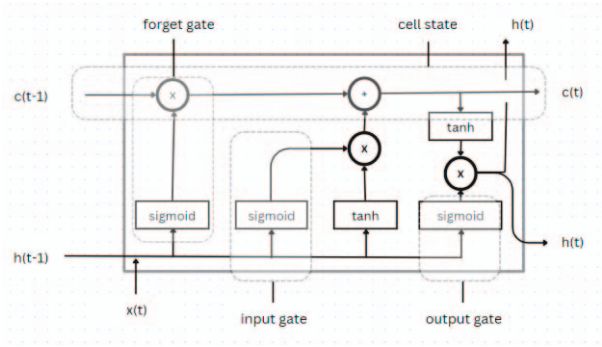

Fig. 3. Long Short-Term Memory Cell

Bidirectional long short-term memory (Bi-LSTM) is the process of making any neural network have the sequence information in both directions, backward (future to past) or forward (past to future). Bidirectional LSTMs have two recurrent components: a forward recurrent component and a backward recurrent component. The forward component computes the hidden and cell states like a standard unidirectional LSTM. In contrast, the backward component computes them by taking the input sequence in a reverse-chronological order, i.e., starting from time step tx to 1. The intuition of using a backward component is that we are creating a way where the network sees future data and learns its weights accordingly to help the network capture some dependencies that otherwise would not have been captured by the standard (forward) LSTM.

d) Word Embeddings

A word embedding is a learned representation of text where words with the same meaning have similar representations. Each dimension of an embedding represents some semantic attribute of the associated word. The main benefit of word embeddings is that they allow machine-learning models to generalize well to texts with words not seen in the training dataset since if those new words have similar meanings to previously seen words, their embeddings[26] have values close to those of the embeddings of the seen words. Word embeddings have traditionally been used as inputs to deep CNNs, wherein the tweet text is input as a matrix of word embeddings.

e) Model Description

The individual usage of Convolutional neural networks (CNNs) [9] and recurrent neural networks (RNNs) have fetched the best results in classification problems. We are leveraging both the architectures, i.e., CNN and RNN, and building a model consisting of a Convolutional neural network where the input will be fed. The output from the CNN will be fed into a Recurrent neural network, especially a Long short-term memory network. Output from the LSTM network will be fed into a fully connected dense layer with SoftMax as an activation function to output the target class label. Inputs to the Convolutional neural network will be word embeddings. We have experimented with word2vec[5], which is pre-trained on crisis-related datasets, glove[26], BERT[27], and Elmo[28] embeddings. A deep-learning CNN model will pass this feature matrix through convolution layers with filters (Kernels), RELU activation, and Pooling layers.

The developed model first uses a convolutional neural network to extract multiple sets of features of the input crisis messages, pool them separately to extract important tweet/message features, and then concatenates the extracted features as input to the LSTM neural network and outputs classification results through the fully connected dense layer. Thus, the developed model includes an input layer, a word embedding layer, a convolution layer, a pooling layer, a bidirectional LSTM network layer, and a fully connected layer with an activation function as SoftMax for multi-class classification. The developed model architecture is briefly described in Fig. 4.

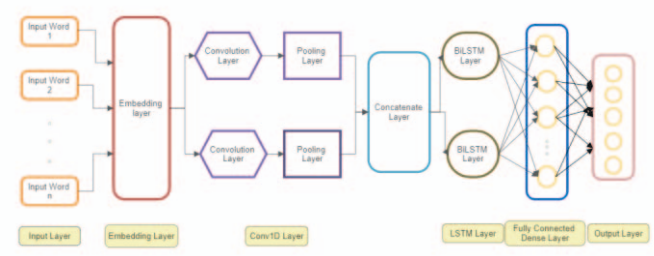

Fig. 4. Developed model architecture

IV. EXPERIMENTS

We have implemented deep learning models using the TensorFlow Keras library and conducted experiments in Google Colaboratory (Google Colab) GPU with a high RAM environment and a complete cloud-based Jupyter notebook environment. Colab is a low-cost notebook that supports popular libraries and does not require any setup.

A. Datasets

We have used disaster messages posted in Hugging Face [16], which contains nearly 30,000 messages captured from different events. The dataset has been labeled with 36 different categories related to disaster response. We have reclassified the messages based on 36 different categories into 5 class labels based on urgency. After applying data augmentation techniques to the minority classes in the training dataset, the number of messages for each class label is shown in Table. I.

TABLE I. Distribution of messages over training and testing

Class Label	Class Title	Number of Messages		
		Train	Val	Test
0	Not Related	4877	610	609
1	Related to disaster but not related to urgency	7833	979	980
2	Related to Urgent AID needed(Basic survival goods immediately required),	6224	778	778
3	Related to Urgent AID needed(Medical Help or Basic things for survival)	5184	648	648
4	Related to Urgent AID needed(Regarding security or military or rescue operations, Missing people, Other).	7403	925	926
Total number of messages		31521	3940	3941

B. Deep Learning Models

After pre-processing the dataset, we padded the data sequences to a standard length. As 99% of messages are within 40 words, setting the same standard length for padding and then splitting the dataset into a train, validation, and test sets in a way that preserves the exact proportions of examples in each class as observed in the original dataset using scikit-learn[24] by mentioning stratify option. We are performing tokenization using the Keras pre-processing library. We have conducted experiments using various word embeddings of 300 dimensions, such as CrisisNLP word embeddings[5], which are pre-trained on the crisis dataset GloVe, BERT, and Elmo. We have also experimented with varied deep learning model architectures, such as hierarchical using CNN/BiLSTM/CBiLSTM and ensemble setting(CNN, BiLSTM). We have also investigated the effectiveness of mentioned pre-trained models, such as BERT[27] from Hugging Face and DistilBERT[30]. The bert-base-uncased model uses 12 layers of transformers block with a hidden size of 768, self-attention heads as 12 and has around 110M trainable parameters, and distilbert-base-uncased has 6-layers, 768-hidden, 12-heads, 66M parameters and DistilBERT model distilled from the bert-base-uncased model checkpoint.

C. Evaluation Metrics

The performance of the developed model has been evaluated by calculating precision, recall, and F1-measure. Here, we cannot only depend on accuracy because of dataset imbalance; a model with high accuracy is not necessarily desirable since it could achieve high accuracy by merely classifying every tweet as a majority class label. Here, precision and recall are particularly relevant because we need to find out what fraction of the few specific "urgent" class label examples the model correctly detects and what fraction of the messages classified as specific "urgent" class labels are, in fact, urgent. Let TP (true positive) denote the number of correct specific urgent classifications, TN (true negative) denote the number of correct not specific urgent classifications, FP (false positive) denote the number of messages incorrectly classified as a specific urgent class, and FN (false negative) denotes the number of messages incorrectly classified as a not specific urgent class. Precision is defined as

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

Recall is defined as

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

F1 Measure is calculated based on precision and recall and is defined as the harmonic mean of precision and recall. It is given as

$$F = \frac{2*Precision*Recall}{Precision+Recall} \quad (6)$$

D. Results

We have conducted experiments using the different model architectures mentioned in Table. III. Table. II mentions configurations considered for model architecture with the highest accuracy.

Table. II. Configuration for a model with the highest accuracy

Type of Layer	**Layer Configuration**	**Activation Function**
Embedding	Pretrained word2vec embeddings of 300 D	N/A
CNN 1D	filters =128, kernel_size=3, strides=2, padding=same,	relu
Maxpooling1D	Pool_size = 2	N/A
CNN 1D	filters =128, kernel_size=3, strides=2, padding=same,	relu
Maxpooling1D	Pool_size = 2	N/A
Bi-directional LSTM	units=128, return_sequences=true, dropout=0.2, recurrent_dropout=0.2	tanh
Dense	units = 128	relu
Dense	units = 64	relu
Dense	Units = 5	softmax

The model with the highest accuracy has total training parameters of 11,064,097 and used Adam optimizer with a learning rate of 0.001, sparse categorical cross entropy as loss function, batch size of 32, trained models with 50 epochs mentioning early-stopping, and used 10-fold cross-validation to reduce overfitting of the model. Table. III contains sparse categorical accuracy, precision, recall, and F1 score calculated on five class labels for each developed model architecture, whereas Table. IV contains accuracy, precision, recall, and F1 score for fine-tuned pre-trained models.

Table. III. Results of experiments using different model architecture

Model Architecture	**Accuracy**	**Precision(%)**	**Recall(%)**	**F1 score(%)**
CNN(2), CONCAT ,BiLSTM(1)	68%	68.2	68.5	68.1
CNN(2), CONCAT, BiLSTM(2)	68%	67.7	68.9	68.1
CNN(4), MaxPooling(2), CONCAT, LSTM(1)	68%	68	68.8	68.2
Stacked BiLSTM(2), Stacked LSTM(2)	68%	68.4	69	68.4
Stacked BiLSTM(4)	68%	67.4	68.3	67.5
CNN(2), MaxPooling(2)	63%	65.2	62.9	62.6
CNN(6), MaxPooling(1)	64%	64.6	64.1	64.1
StackedCNN(2), MaxPooling(2), BiLSTM(2), LSTM(2)	63%	66.2	63	62.9
Hierarchical LSTM(1)	66%	65.5	65.7	65.6
Ensemble CNN(2),BiLSTM(1)	66%	66.1	65.9	65.9

Table. IV. Results of experiments using pre-trained model architecture

Model Architecture	**Accuracy**	**Precision(%)**	**Recall(%)**	**F1 score(%)**
Bert-base-uncased	71%	71.9	69.6	70.2
distilbert-base-uncased	70%	70.9	70.2	70.3

Table. III shows that concatenated CNN followed by BiLSTM gives 68% accuracy results comparable to stacked BiLSTM and LSTM. The fine-tuned pre-trained BERT and DistilBERT models show an accuracy of 71% and 70% and have trainable parameters of 110M and 66M, respectively. Our developed models are performing better even with 11M trainable parameters, and the training time of the model is also less compared with pre-trained models. Table. V briefly describes class label-wise precision, recall, and F1 scores obtained from the model, i.e., concatenated CNN followed by BiLSTM.

Table. V. Precision, Recall, F1 scores for class labels

Class Label	**Precision(%)**	**Recall(%)**	**F1-Score(%)**
Not Related	59	68	63
Related to disaster but not related to urgency	59	49	54
Related to Urgent AID needed(Basic survival goods immediately required),	84	87	86
Related to Urgent AID	71	61	66

needed(Medical Help or Basic things for survival)			
Related to Urgent AID needed (Regarding security or military or rescue operations, Missing people, Other)	68	78	72

V. Conclusion & Future Work

Text and document classification has grown more advanced using natural language processing techniques. In this project, we have gathered a dataset with various disasters and tried classifying various urgency labels, unlike binary urgency classification. We examined the performance of developed model architectures with various word embeddings. Models with concatenated CNN followed by Bidirectional LSTM and stacked BiLSTM outperform other developed model architectures like baseline Multilayer perceptron and CNN models and observed improvement compared to the ensemble (CNN, BiLSTM) or hierarchical models. We have also investigated the performance of pre-trained models such as BERT and Distill BERT, which are fine-tuned over the disaster messages dataset. We have also observed comparable results of using general word embeddings over domain-specific word embeddings. We have concluded that developed models with only 10% of trainable parameters perform better than fine-tuned pre-trained models and have a low training time. We can easily integrate the model into mobile devices as it is not memory intensive. We built a user interface using the Flask web framework for better user interaction with the model. As the dataset covers wide disasters, future experiments will be done to improve the model performance over various urgency labels.

References

[1] E. Kouloumpis, T. Wilson, J. Moore, Twitter Sentiment Analysis: The Good the Bad and the OMG!, in: Fifth International Association for the Advancement of Artificial Intelligence Conference on Weblogs and Social Media, Barcelona, Spain, AAAI Press, 2011, pp. 538–541.

[2] C. dos Santos, M. Gatti, Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts, in: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 69–78.

[3] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: AAAI Conference on Artificial Intelligence, Austin, Texas, United States, Association for Computing Machinery, 2015.

[4] G. Burel, H. Saif, M. Fernandez, H. Alani, On Semantics and Deep Learning for Event Detection in Crisis Situations, in: Workshop on Semantic Deep Learning (SemDeep), at the European Semantic Web Conference (ESWC) 2017, Elsevier, 2017.

[5] M. Imran, P. Mitra, C. Castillo, Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages, in: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), European Language Resources Association (ELRA), Portoroˇz, Slovenia, 2016.

[6] L. Derczynski, K. Meesters, K. Bontcheva, D. Maynard, Helping Crisis Responders Find the Informative Needle in the Tweet Haystack, in: ISCRAM 2018 Conference Proceedings - 15th International Conference on Information Systems for Crisis Response and Management, Rochester, New York, United States, Information Systems for Crisis Response And Management, 2018, pp. 649–662.

[7] V.K. Neppalli, C. Caragea, A. Squicciarini, A. Tapia, S. Stehle, Sentiment analysis during Hurricane Sandy in emergency response, Int. J. Disaster Risk Reduct. 21 (2017) 213–222.

[8] Firoj Alam, Hassan Sajjad, Muhammad Imran and Ferda Ofli, CrisisBench: Benchmarking Crisis-related Social Media Datasets for Humanitarian Information Processing, In ICWSM, 2021.

[9] Devaraj, A.; Murthy, D.; Dontula, A. Machine-learning methods for identifying social media-based requests for urgent help during hurricanes. Int. J. Disaster Risk Reduct. 2020, 51, 101757.

[10] R. Wang, Z. Li, J. Cao, T. Chen and L. Wang, "Convolutional Recurrent Neural Networks for Text Classification," 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1-6, doi: 10.1109/IJCNN.2019.8852406.

[11] Aytuğ Onan,Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification, Journal of King Saud University - Computer and Information Sciences, Volume 34, Issue 5,2022,Pages 2098-2117,ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2022.02.025.

[12] Kruspe, A., Kersten, J., and Klan, F.: Review article: Detection of actionable tweets in crisis events, Nat. Hazards Earth Syst. Sci., 21, 1825–1845, https://doi.org/10.5194/nhess-21-1825-2021, 2021.

[13] Kejriwal M, Zhou P. On detecting urgency in short crisis messages using minimal supervision and transfer learning. Soc Netw Anal Min. 2020;10(1):58. doi: 10.1007/s13278-020-00670-7. Epub 2020 Jul 8. PMID: 32834866; PMCID: PMC7341028.

[14] Efsun Sarioglu Kayi, Linyong Nan, Bohan Qu, Mona Diab, and Kathleen McKeown. 2020. Detecting Urgency Status of Crisis Tweets: A Transfer Learning Approach for Low Resource Languages. In Proceedings of the 28th International Conference on Computational Linguistics, pages 4693–4703, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[15] A. Hassan and A. Mahmood, "Convolutional Recurrent Deep Learning Model for Sentence Classification," in IEEE Access, vol. 6, pp. 13949-13957, 2018, doi: 10.1109/ACCESS.2018.2814818.

[16] https://huggingface.co/datasets/disaster_response_messages

[17] https://github.com/hetpandya/textgenie

[18]**https://towardsdatascience.com/data-augmentation-library-for-text-9661736b13ff**

[19] X. Zhang, J. Zhao and Y. LeCun. Character-level Convolutional Networks for Text Classification. 2015

[20] W. Y. Wang and D. Yang. That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets. 2015

[21] S. Kobayashi. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relation. 2018

[22] C. Coulombe. Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs. 2018

[23] https://flask.palletsprojects.com/en/2.2.x/

[24] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[25] "Convolutional Neural Networks for Sentence Classification" Y. Kim 2014 in Conference on Empirical Methods in Natural Language Processing (EMNLP'14)

[26] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. [pdf] [bib]

[27] Devlin, Jacob, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." ArXiv abs/1810.04805 (2019): n. pag.

[28] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. "Deep contextualized word representations". https://arxiv.org/abs/1802.05365

[29]**https://www.davidsbatista.net/blog/2018/03/31/SentenceClassificationConvNets/**

[30] Victor Sanh and Lysandre Debut and Julien Chaumond and Thomas Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". https://arxiv.org/abs/ 1910.01108

[31] A. Olteanu, S. Vieweg, C. Castillo. 2015. What to Expect When the Unexpected Happens: Social Media Communications Across Crises. In Proceedings of the ACM 2015 Conference on Computer Supported Cooperative Work and Social Computing (CSCW '15). ACM, Vancouver, BC, Canada.

[32] A. Olteanu, C. Castillo, F. Diaz, S. Vieweg. 2014. CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises. In Proceedings of the AAAI Conference on Weblogs and Social Media (ICWSM'14). AAAI Press, Ann Arbor, MI, USA.

[33] Firoj Alam, Ferda Ofli, and Muhammad Imran, CrisisMMD: Multimodal Twitter Datasets from Natural Disasters, In Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM), 2018, Stanford, California, USA. [Bibtex]