Machine Learning with Applications 7 (2022) 100253

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Comparative analysis of contextual and context-free embeddings in disaster prediction from Twitter data

Sumona Deb [a], Ashis Kumar Chanda [b,c,*]

[a] Department of Computer Science and Engineering, Metropolitan University, Bangladesh
[b] Department of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh
[c] Department of Computer and Information Sciences, Temple University, PA, USA

ARTICLE INFO

Keywords:
Social media data
Twitter data
Disaster prediction
BERT
Kaggle competition
Natural Language Processing (NLP)

ABSTRACT

Twitter is a social media site where people post their personal experiences, opinions, and news. Due to the ubiquitous real-time data availability, many rescue agencies monitor this data regularly to identify disasters, reduce risk, and save lives. However, it is impossible for humans to manually check the mass amount of data and identify disasters in real-time. For this purpose, many research have been proposed to present words in machine-understandable representations and apply machine learning methods on the word representations to identify the sentiment of a text. The previous research methods provide a single vector representation or embedding of a word from a given document. However, the recent advanced contextual embedding method (BERT — Bidirectional Encoder Representations from Transformers) constructs different vectors for the same word in different contexts. The BERT embeddings have been used successfully in various Natural Language Processing (NLP) tasks, yet there is no concrete analysis of how these representations are helpful in disaster-type tweet analysis. This research study explores the efficacy of the BERT embeddings on predicting disaster from Twitter data and compares these to traditional context-free word embedding methods. We provide both quantitative and qualitative results for this study. The results show that the contextual embeddings have the best results in disaster prediction task than the traditional word embeddings. Furthermore, we discuss the opportunities and challenges of contextual embeddings on sentiment analysis of Twitter data.

1. Introduction

The rapid growth of the internet in the current decade has made social media as one of the most popular platforms for sharing news and communicating with all people. People regularly post their personal experiences, current events, and local and global news on social media. For this reason, the daily usages of social media has grown and becoming a large dataset. The large dataset becomes an important source of data to make different types of research analysis, such as relation extraction (Ritter et al., 2015), knowledge extraction (Mai et al., 2020) and sentiment analysis (Deho et al., 2018). Moreover, the social media data are real-time data and accessible to monitor. Therefore, several research are conducted to perform different types of real-time predictions using the social media data, such as stock movement prediction (Nguyen et al., 2015), traffic incident detection (Gu et al., 2016), and natural disaster prediction (Yoo et al., 2018).

Twitter is a social media site that can be accessed through people's laptops and smartphones. The rapid growth of smartphone or laptop usages enables people to share an emergency that they observe in real time. For this reason, many disaster relief organizations and news agencies are interested in monitoring Twitter data programmatically. However, unlike long articles, tweets are short length text, and they tend to have more challenges due to their shortness, sparsity (i.e., diverse word content) (Chen et al., 2011), velocity (rapid growth of short text like SMS and tweet) and misspelling (Alsmadi & Gan, 2019). For these reasons, it is very challenging to understand whether a person's words are announcing a disaster or not. For example, a tweet like this, "$\#old Band\ amazing\ performance!\ light,\ color,\ fire\ on\ stage!\ lots\ of\ people\ and\ chaos!$" tells us an experience of a person in a concert and we can say from that he enjoyed it, because of the word, "$amazing$". Even though it contains the word, "$fire$", it does not mean any danger or emergency; rather, it is used to describe the colorful decoration of the stage. Let us assume another tweet like this, "$California\ Hwy.\ 20\ closed\ in\ both\ directions\ due\ to\ Lake\ Country\ fire$". Here, the word "$fire$" means disaster, and the tweet describes an emergency. The two examples show that one word could have multiple meanings based on its context. Therefore, understanding the context of words is important to analyze a tweet's sentiment.

[*] Corresponding author at: Department of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh.
E-mail address: ashis@temple.edu (A.K. Chanda).

https://doi.org/10.1016/j.mlwa.2022.100253
Received 4 October 2021; Received in revised form 26 December 2021; Accepted 7 January 2022
Available online 13 January 2022

Table 1
A summary work list of previous research on twitter data analysis.

Authors	Proposed methods
Palshikar et al. (2018)	A weakly supervised model with BOW representations
Singh et al. (2019)	Markov-based model to predict the location of Tweets during a disaster
Karami et al. (2020)	Text mining methods for twitter situational awareness
Bhoi et al. (2020)	A hybrid model to analyze the emergency-related tweets
Algur and Venugopal (2021)	Naive Bayes, Logistic Regression, Random Forest, and SVM methods for classifying disaster-type tweets
Pota et al. (2021)	A pre-processing method for BERT-based sentiment analysis of tweets

Different researchers proposed different methods to understand the meaning of a word by representing it in embedding or vector (Bojanowski et al., 2016; Mikolov et al., 2013; Pennington et al., 2014). Neural network-based methods such as Skip-gram (Mikolov et al., 2013), FastText (Bojanowski et al., 2016) are popular for learning word embeddings from large word corpus and have been used for solving different types of NLP tasks. These methods are also used for sentiment analysis of Twitter data (Deho et al., 2018; Poornima & Priya, 2020). However, those embedding learning methods provide static embedding for a single word in a document. Hence, the meaning of the word,"$fire$" would remain the same in the above two examples for these methods.

To handle this problem, the authors of Devlin et al. (2018) proposed a contextual embedding learning model, Bidirectional Encoder Representations from Transformers (BERT), that provides embeddings of a word based on its context words. In different types of NLP tasks such as text classification (Sun et al., 2019), text summarization (Liu & Lapata, 2019), entity recognition (Hakala & Pyysalo, 2019), BERT model outperformed traditional embedding learning models. However, it is interesting to discover how the contextual embeddings could help to understand disaster-type texts. For this reason, we plan to analyze the disaster prediction task from Twitter data using both context-free and contextual embeddings in this study. We use traditional machine learning methods and neural network models for the prediction task where the word embeddings are used as input to the models. We show that contextual embeddings work better in predicting disaster-type tweets than the other word embeddings. Finally, we provide an extensive discussion to analyze the results.

The main contributions of this paper are summarized as follows.

1. We analyze a real-life natural language online social network dataset, Twitter data, to identify challenges in human sentiment analysis for disaster-type tweet prediction.
2. We apply both contextual and context-free embeddings in tweet representations for disaster prediction through machine learning methods and show that context-free embeddings (BERT) can improve the accuracy of disaster prediction compared with contextual embeddings.
3. We provide a detailed discussion on the comparison of contextual and context-free embeddings; and the opportunities and challenges of context-free embeddings in the sentiment analysis task. We share our codes publicly that will enable researchers to run our experiments and reproduce our results for future research directions.[1]

The rest of the paper is organized as follows. In Section 2, some related works are introduced. The main methodology of this paper is elaborated in Section 3. The dataset and the experiments are presented in Sections 4 and 5, respectively. Finally, the conclusion is drawn in Section 6.

2. Related work

Many research analyzed Twitter data for understanding emergency situation and predicting disaster analysis (Ashktorab et al., 2014; Karami et al., 2020; Olteanu et al., 2014; Zou et al., 2018). One group of researchers focused on clustering text data to identify a group of tweets that belong to disaster (Ashktorab et al., 2014; Olteanu et al., 2014), another group of researchers used text mining and statistical approaches to understand crises (Karami et al., 2020; Zou et al., 2018). Researchers also proposed different traditional machine learning models to analyze Twitter data and predict disaster or emergency situations where words of a tweet are represented as embeddings (Algur & Venugopal, 2021; Palshikar et al., 2018; Singh et al., 2019). For example, Palshikar et al. (2018) proposed a weakly supervised model where words are presented with a bag of words (BOW) model. The authors in Singh et al. (2019) used a Markov-based model to predict the location of Tweets during a disaster. Bhoi et al. (2020) proposed a hybrid model which is a mix of LSTM (Long Short Term Memory) and CNN (Convolutional Neural Network) based on word embedding to analyze the emergency-related tweets and to classify them. Moreover, frequency-based word representation is used in Algur and Venugopal (2021) for disaster prediction from Twitter data using Naive Bayes, Logistic Regression, Random Forest, and SVM (Support Vector Machine) methods. In a recent work (Pota et al., 2021), the authors proposed a pre-processing method for BERT-based sentiment analysis of tweets. However, it is interesting to explore machine learning model performance on different word embeddings to observe how the context words help to predict a tweet as a disaster. A summary list of the recent works are shown in Table 1.

3. Methodology

In this section, we discuss our approach of leveraging word embedding for disaster prediction from Twitter data using machine learning methods. We consider three types of word embeddings, (1) bag of words (BOW), (2) context-free, and (3) contextual embeddings. The word embeddings are used in both traditional machine learning methods and deep learning models as input for disaster prediction. The Fig. 1 presents the flow of works with a sample example tweet.

3.1. BOW representation

The bag-of-words (BOW) model is a common approach for text representation of a word document. If there are V words in a text vocabulary, then BOW is a binary vector or array of length |V| where each index of the array is used to present one word of the vocabulary. If a word exists in a document, then the corresponding array index of the word becomes one; otherwise, it contains zero. We use BOW embeddings of Twitter data in three traditional machine learning methods such as decision tree, random forest, and logistic regression to predict the sentiment of a tweet.

Even though BOW is good for representing words of a document, it loses contextual information because the order of words is not recorded

[1] https://github.com/ashischanda/sentiment-analysis

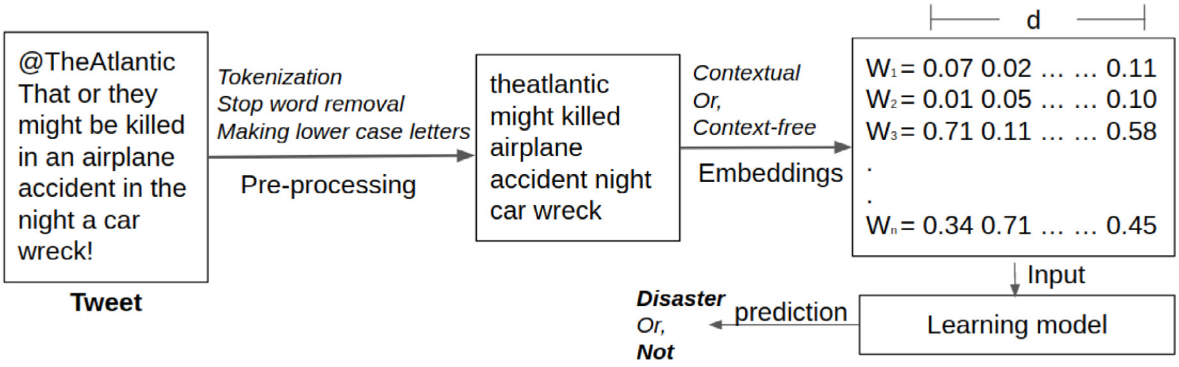


Fig. 1. The overall scheme of the disaster prediction model from Twitter data.

in the binary structure. However, contextual information is required to understand and analyze the sentiment of a text. For this reason, we also plan to use context-based embeddings for this sentiment analysis task.

3.2. Context-free embeddings

Many existing research proposed to learn word embeddings based on the co-occurrences of word pairs in documents. GloVe (Pennington et al., 2014) is one common method for learning word embeddings from the co-occurrences of words in documents. This model is trained on the non-zero values of a global word-word co-occurrence matrix, which presents the frequency of words with one another in a given document. However, neural network-based models such as Skip-gram (Mikolov et al., 2013), FastText (Bojanowski et al., 2016) have became popular recently to learn word representations from documents and used for sentiment analysis.

Skip-gram model scans each word of a given corpus, and learns the probability of observing the surrounding words for the scanned word. Suppose, we have a word sequence, S = {... w_{t-2}, w_{t-1}, w_t, w_{t+1}, w_{t+2}, …} in a corpus, and the scanned word is w_t. Then, the probability of observing surrounding w_i for w_t is defined as

$$\mathcal{L} = \sum_{w_i \in C_{w_t}} \log p(w_i|w_t), \tag{1}$$

where $w_i \in C_{w_t} = (w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ for the context size of two.

FastText also scans word of a given corpus to learn the probability of observing the surrounding words. However, the scanned or target word is represented as a n-grams. The authors (Bojanowski et al., 2016) found that FastText is useful for finding representations of rare words.

In our research study, we use the pre-trained embeddings of three context-free embedding models (GloVe, Skip-gram, FastText) in a neural network-based model to analyze the sentiment of tweet data and predict disaster-type tweets. To represent a tweet in context-free embeddings, we took the average of word embeddings of a tweet following the same strategy of Kenter et al. (2016). For the calculated vector of a tweet, we use softmax to predict the sentiment of the tweet. Let us suppose that the vector of a tweet is x, we have a set of labels, L = {"positive", "negative"} and $Z \in \mathbb{R}^{(|L| \times d)}$ is the weight matrix of softmax function where d is the dimension of feature of vector x. Then, the probability of the tweet to be positive or disaster is calculated as follows

$$p(y(l_i) = 1) = \frac{e^{Z_i \cdot x}}{\sum_{l_k \in L} e^{Z_k \cdot x}} \tag{2}$$

Recently, deep neural networks are also used for sentiment analysis. Different types of Deep neural networks (DNNs) are widely explored to handle different NLP tasks. Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are the two main types of DNN architectures. CNN is supposed to be good at extracting position invariant features while RNN works at modeling units in word sequence (Yin et al., 2017). Previous researchers (Yin et al., 2017) found that RNN is better than CNN in sentiment analysis when the sentiment is determined by the entire sentence and the length of the sentence is short. In our Twitter dataset, we have seen that the average word length of tweets is 12.50. Therefore, we plan to use a bidirectional recurrent neural network with Long Short-Term Memory (LSTM) gates (Hochreiter & Schmidhuber, 1997) to observe how the context-free embeddings work on deep neural networks.

LSTM keeps feedback connections with previous states while Bidirectional-LSTM (Bi-LSTM) has connections on both directions. Bi-LSTM model processes the input words of a tweet from right to left and in reverse. Fig. 2 shows a basic LSTM unit and Bi-LSTM model. The Bi-LSTM block is followed by a fully connected layer with sigmoid function as an activation function for the output layer.

Suppose, we have a tweet of n words and $[w_1, w_2, w_3, \ldots w_n]$ is a list of vector where w_t represents an embedding or vector of word in the given tweet and $1 <= t <= n$. In Fig. 2, w_t is used as an input to the model. In the LSTM unit of Fig. 2, h_t presents hidden state vector or output vector and c_t means cell state vector. A detail of LSTM model could be found in Hochreiter and Schmidhuber (1997).

3.3. Contextual embeddings

Unlike the other word embeddings, BERT model (Devlin et al., 2018) generates different vectors for the same word in different contexts. For example, context-free models such as GloVe and word2vec generate a single $word\ embedding$ representation for the word "$bank$" that could be used in different sentences with different meanings (i.e., "$bank\ deposit$" and "$river\ bank$"). Contextual models such as BERT instead generate a representation of each word that is based on the other words in the sentence.

BERT is a Transformer based language model with multiple numbers of encoder layers and self-attention heads. The model architecture is similar to the original Transformer model (Vaswani et al., 2017). In the language modeling, 15% of tokens were masked and BERT was trained to predict them from context. After completing the training process, BERT learns contextual embeddings for words.

Recent advances in NLP has shown that BERT model has outperformed traditional embeddings in different NLP tasks, like entity extraction, next sentence prediction. In our study, we plan to investigate how well do contextual embeddings work better than traditional embeddings in sentiment analysis. For this purpose, we use the pre-trained embeddings of BERT models in the same neural network models to predict disaster-type tweets.

4. Dataset

For this study, we used a Twitter dataset from a recent Kaggle competition (Natural Language Processing with Disaster Tweets[2]). Kaggle competition is a very well-known platform for machine learning

[2] https://www.kaggle.com/c/nlp-getting-started

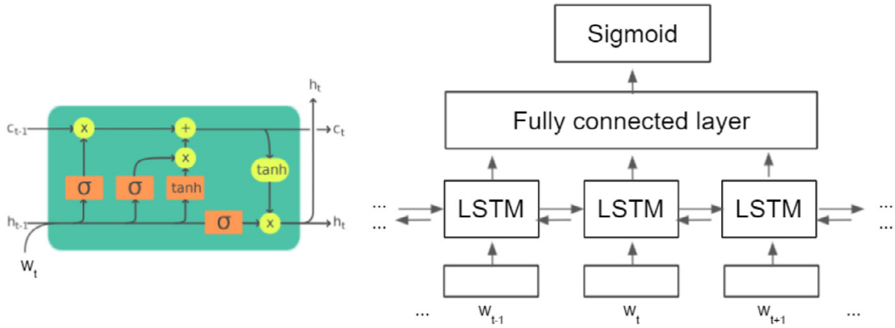


Fig. 2. Showing LSTM unit (at left) and Bi-LSTM model (at right).

Table 2
Sample pre-processed tweets.

Tweet (original)	Tweet (after preprocessing)
#RockyFire Update => California Hwy. 20 closed in both directions due to Lake County fire - #CAfire #wildfires	rockyfire update california hwy 20 closed directions due lake county fire cafire wildfires
@TheAtlantic That or they might be killed in an airplane accident in the night a car wreck!	theatlantic might killed airplane accident night car wreck

Table 3
Training data statistic.

Total train data	7,613
Total positive data (or disaster tweets)	3,271
Total unique words	21,940
Total unique words with frequency > 1	6,816
Avg. length of tweets	12.5
Median length of tweets	13
Maximum length of tweets	29
Minimum length of tweets	1

researchers where many research agencies share their data to solve different types of research problems. For example, many researchers used data from Kaggle competitions to analyze real-life problems and propose models to solve the problems, such as sentiment analysis, feature detection, diagnosis prediction (Iglovikov et al., 2017; Koumpouri et al., 2015; Tolkachev et al., 2020; Yang & Ding, 2020; Yang et al., 2018).

In the selected Kaggle competition, a dataset of 10,876 tweets is given to predict which tweets are about real disasters and which ones are not, using machine learning model. This dataset has two separate files, train (7,613 tweets) and test (3,263 tweets) data, where each row of the train data contains ID, natural language text or tweet, and label. The labels are manually annotated by humans. They labeled a tweet as positive or one, if it is about real disaster; otherwise as negative or zero. On the other hand, the test data has ID and natural language text but no label. The competition site stores the labels of test data privately and uses that to calculate test scores based on user's machine learning model predictions and create leader-board for the competition based on the test score. Moreover, this dataset was created by the figure-eight company and originally shared on their website.[3]

We used the training data to train different machine learning models and predict test data labels using trained models. We reported both the train and test data score in our experiment. Note that our purpose is not to get a high score in the competition, rather using Twitter data to study our research goals.

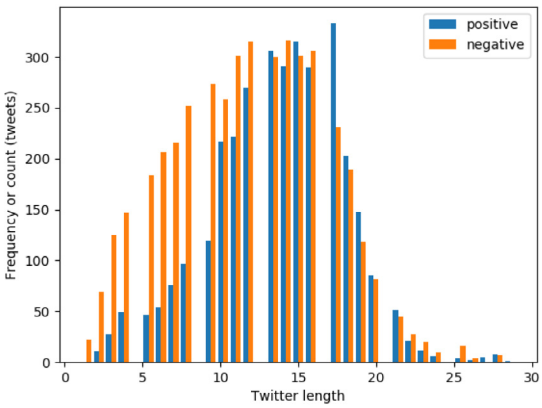


Fig. 3. Twitter word length distribution in training data.

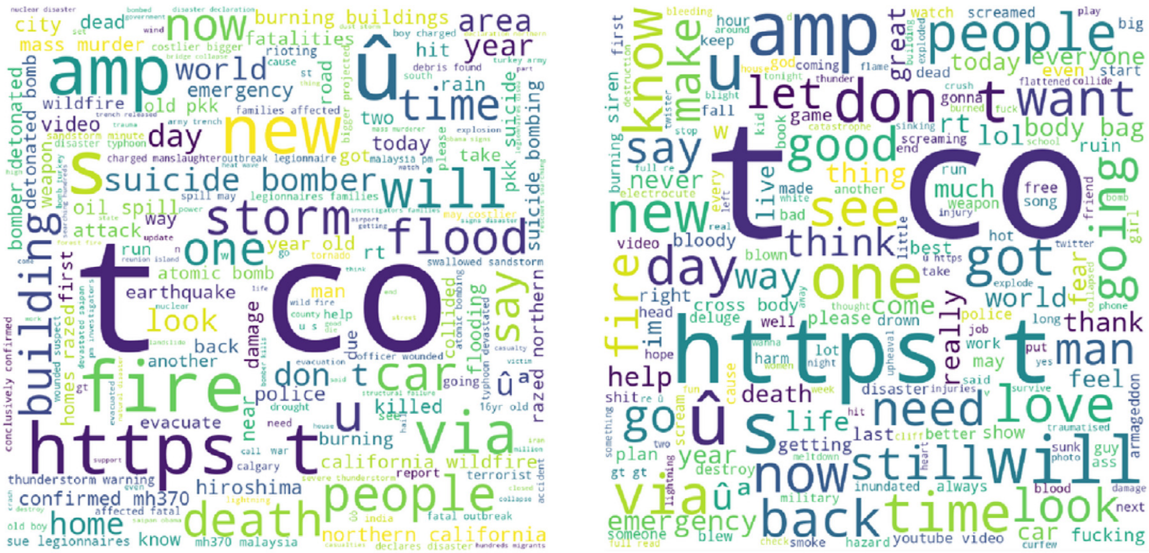
4.1. Data pre-processing

Since the Twitter data is a natural language text, it contains different types of typos, punctuation, abbreviations, and numbers. For this reason, before training machine learning models on the natural language text, a text pre-processing step is required to remove stop words and word tokenization. Hence, we removed all the stopwords and punctuations from the training data and converted all the words into lower-case letters. Table 2 shows some pre-processed tweets with the original tweets.

4.2. Data analysis

Before running any machine learning methods on our data, we analyzed our dataset to obtain some insights about the data. Table 3 shows some statistical results on the training data after pre-processing the text. From the table, we find that there are 43% tweets that are annotated as disaster-type tweets and the other 57% are not disaster-type tweet. There are a total of 21,940 unique words, while only 6,816 words have frequency >1. The average word length of tweets is 12.50. However, it is important to check the length of positive and negative tweets separately to verify whether they have common characteristics.

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



(a) Word cloud for disaster tweets

(b) Word cloud for non-disaster tweets

Fig. 4. Showing the most frequent words in training data.

Fig. 3 shows word distribution for both the positive and negative tweets. The figure shows many negative tweets with small word lengths (<10), but most positive and negative tweets are between a word length of 10 to 20.

We also analyzed the word frequency for positive and negative tweets. Fig. 4 shows the most frequent words in a word cloud where the font size presents the word frequency. If the font size is high for a word, it means the word has high frequency. We can find some common words in both types of tweets (i.e., https, t, co, people). However, Fig. 4(a) highlights many disaster-type words like storm, fire, bomber, death, and earthquake. On the other hand, Fig. 4(b) highlights daily used words such as think, good, love, now, time. From this figure, it is clear that the most frequent words are different in the two types of tweets, and understanding the meaning of words is important for classifying them.

5. Experiments

In our experimental study, we conducted several experiments based on the real Twitter data to predict disaster-type tweets. At first, we describe the experimental settings and the model training procedures in this section. Then, we analyze the experimental results in detail.

5.1. Experimental settings

5.1.1. Traditional ML models with BOW embeddings

From Table 3, we found that the training data has 21,940 unique words where 6816 words have frequency more than 1. To avoid infrequent words, we considered only the vocabulary of 6,816 words in our BOW representations. To represent a tweet in the BOW embeddings, we took a binary array of 6,816 length where it had a value of one for a word which was present in a tweet, otherwise the value was zero.

We used the BOW embeddings to predict the sentiment of a tweet using three traditional machine learning models, (1) decision tree, (2) random forest and (3) logistic regression. We used python Sklearn package[4] and used all the default parameters to train the models on our train dataset. After training the model, we used the test data to get labels and submitted that in Kaggle to have test score.

5.1.2. Deep learning models with context-free embeddings

For this experiment, we chose three context-free methods, (1) Skip-gram (Mikolov et al., 2013), (2) FastText (Bojanowski et al., 2016), and (2) GloVe (Pennington et al., 2014). We used publicly available pre-trained embeddings of Skip-gram and fastText models that are trained on Wikipedia data.[5] The pre-trained embeddings of FastText are collected from Mikolov et al. (2018). The feature size or dimension of all the pre-trained embeddings is 300.

The proposed softmax model is trained for 100 epochs using a stochastic gradient algorithm to minimize the categorical cross entropy loss function. We took 1% of training data as validation data and used the validation data to stop the training model if the loss value for the validation data did not decrease in last ten epochs. Similarly, for the softmax model, we also trained our Bi-LSTM model using the batch gradient descent algorithm for 100 epochs to minimize the binary cross entropy loss function. We followed the same stop rule for this model.

5.1.3. Deep learning models with contextual embeddings

To obtain contextual embeddings, we downloaded publicly available pre-trained BERT model (Devlin et al., 2018) from the official site.[6] The authors proposed several versions of the BERT model, such as BERT-base-uncased, BERT-base-cased, BERT-large-uncased, BERT-large-cased. The base models have 12 hidden layers with 768 vector dimensions, while the large models have 24 hidden layers with 1,024 vector dimensions. Moreover, the text has been lowercased in the uncased model, while in the cased model, the text is the same as the input text (no changes). It means "english" and "English" are the same in the uncased model.

Since we converted all the text in a tweet to lower case letters, we used the BERT-base-uncased model in our experiment. However, we ran another experiment where we did not apply any pre-processing steps on the input text (tweet) and used the BERT-base-cased model to predict disaster-type tweets. We also tried to run the BERT-large-uncased, BERT-large-cased model. However, the models required high computational power (almost 340M parameters). Since we have a limited computational resource, we could not run the models on our dataset.

[4] https://scikit-learn.org/stable/

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

[6] https://github.com/google-research/bert

We gave tweets as inputs in the BERT model and took the hidden states of the $[CLS]$ token of the last layer from the model as embeddings of the given tweets. Then, the embeddings is used in our softmax model to predict sentiment of tweets. The same setting is used in a previous paper (Ji et al., 2021) to predict patient diagnosis from medical note words using pre-trained BERT model.

Moreover, we can find embeddings of each words of a tweet from the pre-trained BERT model. The BERT's pre-trained word embeddings are used as input to our Bi-LSTM model. The authors of Lu et al. (2020) used the similar setting for the sentiment analysis of text data.

5.2. Evaluation metric

Three different metrics are used in our experiment to evaluate the performance of the machine learning models on the disaster prediction task such as, (1) accuracy, (2) F1 score, and (3) Area Under the Curve (AUC). In our experiment, we considered disaster-type tweets as 'positive class' and others as 'negative class'. Hence, True Positive (TP) means the actual disaster tweets that are predicted as disaster while False Positive (FP) shows the tweets that are actually false, but predicted as true. True Negative (TN) and False Negative (FN) imply in the same way. The accuracy is the number of correctly predicted tweets among all the tweets and it is calculated as follows.

$$\text{Accuracy (Acc)} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \tag{3}$$

The F1 score is another popular metric to test predictive performance of a model. It is measured by the harmonic mean of recall and precision where recall means the number of true labels that are predicted by a model among the total number of existing true labels and precision means the number of true labels that are predicted by a model divided by the total number of labels are predicted by the model. The F1 score is calculated as follows.

$$\text{Recall (R)} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4}$$

$$\text{Precision (P)} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5}$$

$$\text{F1 score (F1)} = \frac{2 \times (\mathrm{P} \times \mathrm{R})}{(\mathrm{P} + \mathrm{R})} \tag{6}$$

On the other hand, the AUC tells us how much a model is capable of distinguishing between classes. The higher score of the AUC means the model works better at predicting negative classes as zero and positive classes as one.

5.3. Experimental results

5.3.1. Quantitative results

Table 4 provides the results of all the machine learning models on the disaster prediction tasks for all three types of embeddings. The table shows results for both the training and test data. Since the test data results are collected from the Kaggle competition, we only can report the accuracy score.

The table shows that the logistic regression model has the best results for the BOW embeddings among the three traditional machine learning models. However, the results of neural network models for context-free embeddings are better than the traditional machine learning models that used context-free embeddings as inputs. Among the three context-free embeddings (Skip-gram, FastText, GloVe), the GloVe with Bi-LSTM model has the best train and test score for all the three evaluation metrics. Note that the results also show us that deep learning model like Bi-LSTM has better results than the shallow neural network model such as the softmax model.

Moreover, when we used the same shallow neural network and deep learning models for contextual embeddings such as BERT; we found that there were 2% improvements on AUC and Acc over the context-free embeddings for validation data. It also has 2% improvements on Acc over the context-free embeddings for test data. Moreover, the contextual embeddings with Bi-LSTM have 8% improvements on the F1-score over the context-free embeddings.

Table 4
Performance of different machine learning models on disaster prediction for different types of word representations or embeddings.

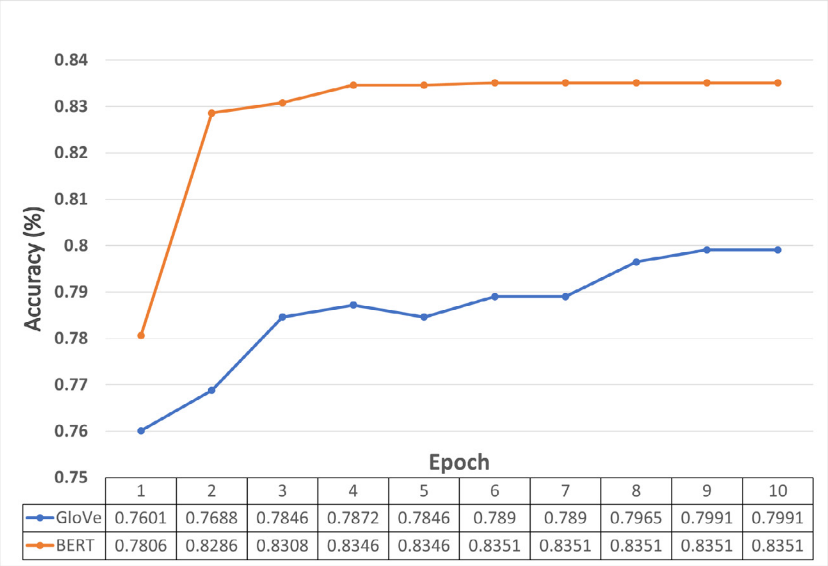
Model	Validation data			Test data
	AUC	F1	Acc	Acc
BOW representation				
Decision tree	0.6320	0.5896	0.6273	0.6380
Random forest	0.8313	0.7320	0.7848	0.7042
Logistic regression	**0.8660**	**0.7443**	**0.7927**	**0.7293**
Context-free embeddings				
Skip-gram+Softmax	0.8281	0.7301	0.7769	0.7649
FastText+Softmax	0.8336	0.7231	0.7769	0.7826
GloVe+Softmax	0.8246	0.7323	0.7717	0.7827
Skip-gram+Bi-LSTM	0.8272	0.7440	0.7808	0.7775
FastText+Bi-LSTM	0.8327	0.7369	0.7817	0.7955
GloVe+Bi-LSTM	**0.8351**	**0.7500**	**0.7991**	**0.8093**
Contextual embeddings				
BERT(cased)+Softmax	0.8392	0.8161	0.8165	0.8203
BERT(uncased)+Softmax	0.8513	0.8254	0.8292	0.8250
BERT(cased)+Bi-LSTM	0.8472	0.8232	0.8254	0.8269
BERT(uncased)+Bi-LSTM	**0.8578**	**0.8316**	**0.8351**	**0.8308**

One interesting point to note is that accuracy is used when the True Positives and True negatives are more important while F1-score is used when the False Negatives and False Positives play a significant role in a problem. Since our problem is predicting disaster-type tweets for saving life and property by warning the disaster relief organizations and news agencies, it is crucial to identify False Negatives and False Positives tweets. Moreover, accuracy can be used when the class distribution is similar. However, the number of disaster-type tweets is not high in real-life as well as in our selected dataset (i.e., 57% of all the tweets are not disaster-type tweets). Hence, F1-score is a better metric for the imbalanced dataset. Therefore, we can conclude that contextual embeddings are helpful and have better performance than the context-free embeddings for the disaster prediction task.

Furthermore, the result of the BERT-base-uncased model is better than BERT-base-cased model. Although BERT-base-cased model has better performance than BERT-base-uncased model on entity extraction, POS (Parts-Of-Speech) tagging (Devlin et al., 2018), we found that BERT-base-uncased model has the best result in our sentiment classification problem (disaster-type tweets prediction problem). The authors of Kula et al. (2020) also found that the BERT-base-uncased model has higher accuracy than the BERT-base-cased model on fake news prediction from news description.

Accuracy over epoch: In Fig. 5, we show the accuracy of Bi-LSTM model for both context-free (i.e., GloVe) and contextual (i.e., BERT) embeddings. From the figure, we observe that the accuracy for both embeddings was low at the beginning and it grows in each epoch. However, the model with BERT reaches at the convergence very quickly (after 6 epochs) than the GloVe. The result shows us that contextual model is faster than context-free method in learning and converges quickly.

Accuracy over tweeter length: We ran another experiment to check how the performance of a model varies over the word length of tweets. For this experiment, we made five groups of tweets from our validation data with different word length ranges, such as 1 to 5, 6 to 10, 11 to 15, 16 to 20 and 21 to 25. Then, we calculated the accuracy of Bi-LSTM model with context-free (GloVe) and contextual (BERT) embeddings for each group of tweets. Fig. 6 shows the result. From the figure, we find that the accuracy was high for both types of embeddings on short tweets and the accuracy decreased when the word



	1	2	3	4	5	6	7	8	9	10
GloVe	0.7601	0.7688	0.7846	0.7872	0.7846	0.789	0.789	0.7965	0.7991	0.7991
BERT	0.7806	0.8286	0.8308	0.8346	0.8346	0.8351	0.8351	0.8351	0.8351	0.8351

Fig. 5. Showing accuracy of Bi-LSTM model for different embeddings on each epoch.

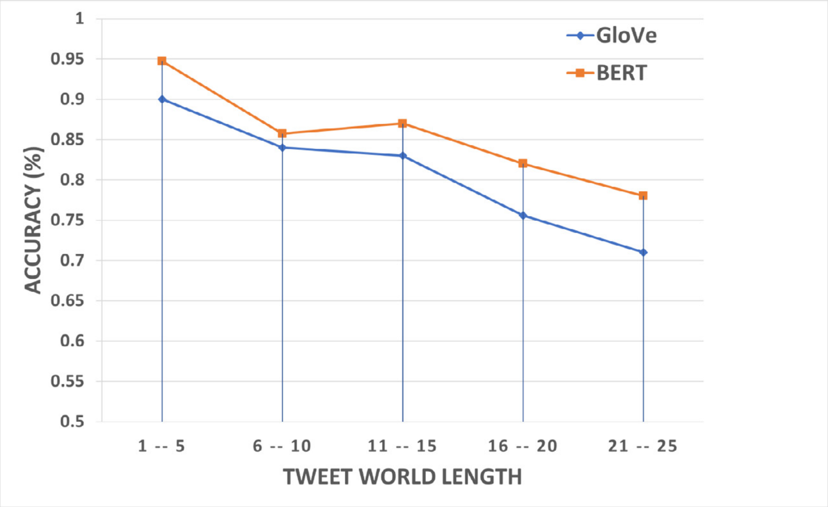


Fig. 6. Showing accuracy of Bi-LSTM model for different embeddings on different tweet word lengths.

length of tweet increased. However, if we check the line graph in Fig. 6, then we can discover that the slope of the GloVe line is higher than the slope of the BERT line. The results inform us that BERT embeddings are useful in predicting disaster-type tweets even if the tweets are longer.

5.3.2. Qualitative results

Table 4 shows us the quantitative results for the prediction of disaster tweets where the neural model with contextual embeddings outperformed the other models. However, it is difficult to understand from the result that when the contextual embeddings predict a disaster tweet successfully while context-free models fail. For this purpose, we observed the prediction results of the Bi-LSTM model for both the context-free (GloVe) and contextual embeddings (BERT). Table 5 shows the model predictions with true labels for some sample tweets. From the table, we can find that the predictions for GloVe embeddings for the first two tweets are positive, maybe because of the word, "*accident*", in the tweets, but the true labels are negative for the two tweets. If we read the tweets, then we can understand that the tweets are not related to disaster or crisis. On the other hand, since the BERT model generates word embeddings based on the context words, it successfully predicted the tweets as negative.

The predictions for the GloVe embeddings for the third and fourth tweets are negative, while their actual labels are positive. Note that no disaster-type words are used in the two tweets, but the tweets described serious situations. The predictions for BERT embeddings are also correct for the third and fourth tweets. The predictions of GloVe and BERT embeddings for the fifth and sixth tweets of Table 4 are correct. Since there are some disaster-type words (i.e., suicide, bomber, bombing) in the tweets, both models successfully labeled them.

After analyzing the results of Table 5, it can be implied that the context-free embeddings are helpful to predict a tweet as a disaster if disaster-type words (i.e., accident, bomb) exist in the tweet. In contrast, contextual embeddings help to understand the context of a tweet that is challenging and important for the sentiment analysis task. Although every tweet has a short length text, contextual embeddings work efficiently to understand the sentiment of a tweet.

6. Discussion

Representing a word as a low-dimensional vector or embedding in sentiment analysis has received significant attention from NLP researchers in the last decade. Researchers proposed both context-free

Table 5
Showing sentiment predictions of Bi-LSTM model for pre-trained GloVe and BERT embeddings [Here, #W means the number of words in a tweet].

	Sample tweets	#W	Prediction		True label
			GloVe	BERT	
1	I swear someone needs to take it away from me, cuase I'm just accident prone.	15	Yes	No	No
2	@Dave if I say that I met her by accident this week- would you be super jelly Dave? :p	19	Yes	No	No
3	Schoolgirl attacked in Seaton Delaval park by 'pack of animals'	10	No	Yes	Yes
4	Not sure how these fire-workers rush into burning buildings but I'm grateful they do. #TrueHeroes	16	No	Yes	Yes
5	A suicide bomber has blown himself up at a mosque in the south	13	Yes	Yes	Yes
6	Bombing of Hiroshima 1945	4	Yes	Yes	Yes

and contextual embedding learning models while the contextual models outperformed context-free models in several downstream tasks. However, there are still many challenges in understanding the sentiment of text, especially for short-length data like tweets.

For example, people use slang words, abbreviations, and text emoticons in tweets to share their experiences. Thus, the pre-processing of Twitter data plays an important role in sentiment analysis. In our experiment, we considered both pre-processed and original tweets as input to the prediction model. We used the BERT-based-uncased model for the pre-processed tweet, while the BERT-based-cased model was used for the original tweet. We found that the model with pre-processed tweet data has the best accuracy. However, different pre-processing steps are discussed in Pota et al. (2021) for BERT-based sentiment analysis of tweets that would be interesting to explore for the disaster-type tweet prediction task.

The length of tweet is another important issue for analyzing sentiment of text. From our study, we observed that the accuracy of disaster-type tweet prediction is high when the tweet length is 1 to 15 and the accuracy dropped when the length increased.

Moreover, transformer-based neural network models such as BERT require a significant amount of memory for training. High computational machine or cloud storage capacity is required to run the models for a large dataset like Twitter. These models have to be compressed to meet the computation and storage constraints.

Although BERT has achieved state-of-the-art results in many NLP tasks, BERT is pre-trained for a generic task (like language modeling), and thus, it lacks domain awareness. In our experiment, we applied machine learning models with BERT in a straightforward manner. However, the authors of Du et al. (2020) proposed a post-training procedure to reduce the gap between in-domain and fully cross-domain environments for the BERT model. The post-training procedure teaches BERT to be domain-aware and extract the domain-specific features in a self-supervised way. We plan to test the post-training process on BERT for our sentiment analysis problem in the future. Moreover, future work should consider replacing BERT with other language models such as RoBERTa (Liu et al., 2019) or XLNet (Yang et al., 2019) in the disaster-type tweet prediction task.

7. Conclusion

In this paper, we compared the performance of pre-trained contextual and context-free embeddings on disaster-type tweet prediction tasks. Our experimental results show that contextual embeddings like BERT's have better F1-score and accuracy than context-free embeddings for predicting disaster from tweets using neural network models. The results also showed that the BERT model has better accuracy results than the context-free embeddings on the variable lengths of tweets. From the qualitative study, we found that traditional context-free embedding methods can detect a tweet as a disaster if disaster-type keywords exist in the tweet, while contextual embeddings can detect disaster-type tweets by understanding the context words. Advanced deep neural network models for text classification (Minaee et al., 2021) such as multilayer convolutional models with attention layers could also be applicable for this prediction task with the pre-trained embeddings to achieve high accuracy.

CRediT authorship contribution statement

Sumona Deb: Conceptualization, Methodology, Data curation, Investigation, Writing – original draft. **Ashis Kumar Chanda:** Conceptualization, Methodology, Investigation, Writing – original draft, review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to express our deep gratitude to the anonymous reviewers of this paper. Their useful comments have played a significant role in improving the quality of this work. We are also thankful to the "Student Success Center" of Temple University for providing kind feedback on the paper writing.

References

Algur, S. P., & Venugopal, S. (2021). Classification of disaster specific tweets-a hybrid approach. In *2021 8th International conference on computing for sustainable global development* INDIACom, (pp. 774–777). IEEE.

Alsmadi, I., & Gan, K. H. (2019). Review of short-text classification. *International Journal Of Web Information Systems*.

Ashktorab, Z., Brown, C., Nandi, M., & Culotta, A. (2014). Tweedr: Mining twitter to inform disaster response. In *ISCRAM* (pp. 269–272). Citeseer.

Bhoi, A., Pujari, S. P., & Balabantaray, R. C. (2020). A deep learning-based social media text analysis framework for disaster resource management. *Social Network Analysis and Mining*, *10*(1), 1–14.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

Chen, M., Jin, X., & Shen, D. (2011). Short text classification improved by learning multi-granularity topics. In *Twenty-second international joint conference on artificial intelligence*. Citeseer.

Deho, B. O., Agangiba, A. W., Aryeh, L. F., & Ansah, A. J. (2018). Sentiment analysis with word embedding. In *2018 IEEE 7th international conference on adaptive science & technology* ICAST, (pp. 1–4). IEEE.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Du, C., Sun, H., Wang, J., Qi, Q., & Liao, J. (2020). Adversarial and domain-aware BERT for cross-domain sentiment analysis. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 4019–4028).

Gu, Y., Qian, Z. S., & Chen, F. (2016). From Twitter to detector: Real-time traffic incident detection using social media data. *Transportation Research Part C (Emerging Technologies)*, *67*, 321–342.

Hakala, K., & Pyysalo, S. (2019). Biomedical named entity recognition with multilingual BERT. In *Proceedings of the 5th workshop on BioNLP open shared tasks* (pp. 56–61).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Iglovikov, V., Mushinskiy, S., & Osin, V. (2017). Satellite imagery feature detection using deep convolutional neural network: A kaggle competition. arXiv preprint arXiv:1706.06169.

Ji, S., Hölttä, M., & Marttinen, P. (2021). Does the magic of BERT apply to medical code assignment? A quantitative study. arXiv preprint arXiv:2103.06511.

Karami, A., Shah, V., Vaezi, R., & Bansal, A. (2020). Twitter speaks: A case of national disaster situational awareness. *Journal of Information Science*, *46*(3), 313–324.

Kenter, T., Borisov, A., & De Rijke, M. (2016). Siamese cbow: Optimizing word embeddings for sentence representations. arXiv preprint arXiv:1606.04640.

Koumpouri, A., Mporas, I., & Megalooikonomou, V. (2015). Evaluation of four approaches for" sentiment analysis on movie reviews" the kaggle competition. In *Proceedings of the 16th international conference on engineering applications of neural networks* INNS, (pp. 1–5).

Kula, S., Choraś, M., & Kozik, R. (2020). Application of the BERT-based architecture in fake news detection. In *Conference on complex, intelligent, and software intensive systems* (pp. 239–249). Springer.

Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Lu, Z., Du, P., & Nie, J.-Y. (2020). VGCN-BERT: augmenting BERT with graph embedding for text classification. *Advances In Information Retrieval*, *12035*, 369.

Mai, M., Leung, C. K., Choi, J. M., & Kwan, L. K. R. (2020). Big data analytics of Twitter data and its application for physician assistants: who is talking about your profession in Twitter? In *Data management and analysis* (pp. 17–32). Springer.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. CoRR, arXiv:1301.3781.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2018). Advances in pre-training distributed word representations. In *Proceedings Of the international conference on language resources and evaluation*. LREC 2018.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning–based text classification: A comprehensive review. *ACM Computing Surveys*, *54*(3), 1–40.

Nguyen, T. H., Shirai, K., & Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems With Applications*, *42*(24), 9603–9611.

Olteanu, A., Castillo, C., Diaz, F., & Vieweg, S. (2014). Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *Eighth international AAAI conference on weblogs and social media*.

Palshikar, G. K., Apte, M., & Pandita, D. (2018). Weakly supervised and online learning of word models for classification to detect disaster reporting tweets. *Information Systems Frontiers*, *20*(5), 949–959.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* EMNLP, (pp. 1532–1543).

Poornima, A., & Priya, K. S. (2020). A comparative sentiment analysis of sentence embedding using machine learning techniques. In *2020 6th International conference on advanced computing and communication systems* ICACCS, (pp. 493–496). IEEE.

Pota, M., Ventura, M., Fujita, H., & Esposito, M. (2021). Multilingual evaluation of pre-processing for BERT-based sentiment analysis of tweets. *Expert Systems With Applications*, *181*, Article 115119.

Ritter, A., Wright, E., Casey, W., & Mitchell, T. (2015). Weakly supervised extraction of computer security events from twitter. In *Proceedings of the 24th international conference on world wide web* (pp. 896–905).

Singh, J. P., Dwivedi, Y. K., Rana, N. P., Kumar, A., & Kapoor, K. K. (2019). Event classification and location prediction from tweets during disasters. *Annals Of Operations Research*, *283*(1), 737–757.

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune bert for text classification? In *China national conference on chinese computational linguistics* (pp. 194–206). Springer.

Tolkachev, A., Sirazitdinov, I., Kholiavchenko, M., Mustafaev, T., & Ibragimov, B. (2020). Deep learning for diagnosis and segmentation of pneumothorax: The results on the kaggle competition and validation against radiologists. *IEEE Journal Of Biomedical And Health Informatics*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances In Neural Information Processing Systems*, *32*.

Yang, X., & Ding, J. (2020). A computational framework for iceberg and ship discrimination: Case study on kaggle competition. *IEEE Access*, *8*, 82320–82327.

Yang, X., Zeng, Z., Teo, S. G., Wang, L., Chandrasekhar, V., & Hoi, S. (2018). Deep learning for practical image recognition: Case study on kaggle competitions. In *Proceedings Of The 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 923–931).

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. arXiv preprint arXiv:1702.01923.

Yoo, S., Song, J., & Jeong, O. (2018). Social media contents based sentiment analysis and prediction system. *Expert Systems With Applications*, *105*, 102–111.

Zou, L., Lam, N. S., Cai, H., & Qiang, Y. (2018). Mining Twitter data for improved understanding of disaster resilience. *Annals Of The American Association Of Geographers*, *108*(5), 1422–1441.