future internet

Article

A Sentiment-Aware Contextual Model for Real-Time Disaster Prediction Using Twitter Data

Guizhe Song * and Degen Huang

School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China; huangdg@dlut.edu.cn
* Correspondence: guizhesong@mail.dlut.edu.cn

Abstract: The massive amount of data generated by social media present a unique opportunity for disaster analysis. As a leading social platform, Twitter generates over 500 million Tweets each day. Due to its real-time characteristic, more agencies employ Twitter to track disaster events to make a speedy rescue plan. However, it is challenging to build an accurate predictive model to identify disaster Tweets, which may lack sufficient context due to the length limit. In addition, disaster Tweets and regular ones can be hard to distinguish because of word ambiguity. In this paper, we propose a sentiment-aware contextual model named SentiBERT-BiLSTM-CNN for disaster detection using Tweets. The proposed learning pipeline consists of SentiBERT that can generate sentimental contextual embeddings from a Tweet, a Bidirectional long short-term memory (BiLSTM) layer with attention, and a 1D convolutional layer for local feature extraction. We conduct extensive experiments to validate certain design choices of the model and compare our model with its peers. Results show that the proposed SentiBERT-BiLSTM-CNN demonstrates superior performance in the F1 score, making it a competitive model in Tweets-based disaster prediction.

Keywords: natural language processing; text classification; mining information; Tweet data; social media

check for updates

Citation: Song, G.; Huang, D. A Sentiment-Aware Contextual Model for Real-Time Disaster Prediction Using Twitter Data. *Future Internet* **2021**, *13*, 163. https://doi.org/10.3390/fi13070163

Academic Editors: Massimo Esposito, Giovanni Luca Masala, Aniello Minutolo and Marco Pota

Received: 21 May 2021
Accepted: 21 June 2021
Published: 25 June 2021

1. Introduction

Social media has been increasingly popular for people to share instant feelings, emotions, opinions, stories, and so on. As a leading social platform, Twitter has gained tremendous popularity since its inception. The latest statistical data show that over 500 million Tweets are sent each day, generating a massive amount of social data that are used by numerous upper-level analytical applications to create additional value. Meanwhile, numerous studies have adopted Twitter data to build natural language processing (NLP) applications such as named entity recognition (NER) [1], relation extraction [2], question and answering (Q&A) [3], sentiment analysis [4], and topic modeling [5].

In addition to the social function, Twitter is also becoming a real-time platform to track events, including accidents, disasters, and emergencies, especially in the era of mobile Internet and 5G communication, where smartphones allow people to post an emergency Tweet instantly online. Timing is the most critical factor in making a rescue plan, and the rise in social media brings a unique opportunity to expedite this process. Due to this convenience, more agencies like disaster relief organizations and news agencies are deploying resources to programmatically monitor Twitter, so that first responders can be dispatched and rescue plans can be made at the earliest time. However, processing social media data and retrieving valuable information for disaster prediction requires a series of operations: (1) perform text classification on each Tweet to predict disasters and emergencies; (2) determine the location of people who need help; (3) calculate the priorities to schedule rescues. Disaster prediction is the first and most important step, because a misclassification may result in a waste of precious resources which could have been dispatched to real needs [6].

However, to automate this process, an accurate and robust classifier is needed to distinguish real disaster Tweets from regular ones. Disaster prediction based on Tweets is challenging, because words indicative of a disaster, such as "fire", "flood", and "collapse", can be used by people metaphorically to describe something else. For example, a Tweet message "On plus side look at the sky last night it was ABLAZE" explicitly uses the word "ABLAZE" but means it metaphorically. The length limit of Tweets brings pros and cons for training a classifier. The benefit is that users are forced to tell a story in a concise way, and the downside is that the lack of clear context may prevent a classifier from well understanding and interpreting the real meaning of a Tweet. Therefore, it is crucial to build an advanced model that can understand the subtle sentiment embedded in Tweets along with their given contexts to make better predictions.

Recent advances in deep learning have explored approaches to address these challenges that are commonly seen in other NLP tasks. Convolutional neural networks (CNNs), which have been widely used in numerous computer vision tasks, have also been successfully applied in NLP systems due to their ability for feature extraction and representation. Recurrent neural networks and their popular variants, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), are not only suitable for general sequential modeling tasks but also provide the capability to capture long dependency information between words in a sentence. In addition, LSTM and GRU can well address the gradient explosion and vanishing issue and allow a training algorithm to converge. Another breakthrough architecture is Bidirectional Encoder Representations from Transformers (BERT), which stacks layers of Transformer encoders with a multi-headed attention mechanism to enhance a model's ability to capture contextual information.

Inspired by these prior efforts, we propose a learning pipeline named SentiBERT-BiLSTM-CNN for disaster prediction based on Tweets. As shown in Figure 1, the pipeline consists of three consecutive modules, including (1) a SentiBERT-based encoder that aims to transform input tokens to sentiment-aware contextual embeddings, (2) a Bidirectional LSTM (BiLSTM) layer with attention to produce attentive hidden states, and (3) a single-layer CNN as a feature extractor. In addition, a standard detection head takes as input a concatenation of the generated features and feeds them into a fully connected layer followed by a softmax layer to output the prediction result, i.e., disaster Tweet or not. The design is validated through extensive experiments, including hyper-parameter tuning to decide certain design choices and an ablation study to justify the necessity of each selected building block. Results show that the proposed system achieves superior performance in the F1 score, making it a competitive model in Tweets-based disaster prediction.

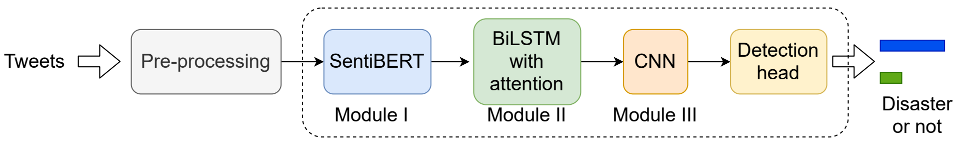


Figure 1. The proposed SentiBERT-BiLSTM-CNN learning pipeline for disaster prediction using Tweets.

The rest of this paper is organized as follows: Section 2 reviews relevant studies; Section 3 covers the dataset description and the technical details of the proposed learning model; Section 4 provides experimental validation with result analysis; Section 5 summarizes our work and points out future directions.

2. Related Work

2.1. Social Media Learning Tasks

Data collected from social media have a lot of potentials to explore. Social texts have been extensively studied and mined to build a wide range of NLP applications such as NER [1], Q&A [3], sentiment analysis [4,7–10], and topic modeling [5,11,12]. In addition, social data have been utilized for emergency, disease, and disaster analysis [13–15]. In [16], the authors develop predictive models to detect Tweets that present situational awareness. The models are evaluated in four real-world datasets, including the Red River floods of 2009 and 2010, the Haiti earthquake of 2010, and the Oklahoma fires of 2009. This paper focuses on exploring the contextual information in Tweets to build a robust disaster classifier.

2.2. RNN/CNN-Based Models in Text Mining

Active development in deep learning in recent years has generated fruitful achievements in social text learning. As two representative learning models, RNN and CNN have been seen in numerous studies, either individually or in a hybrid fashion.

Huang et al. [17] combined BiLSTM and Conditional Random Field (CRF) to build a sequential tagging framework that can be applied to parts of speech (POS), chunking, and NER tasks. In [18], Liu et al. propose a Stochastic Answer Network (SAN) that stacks various layer types, including GRU, BiLSTM, and self-attention; along with a stochastic prediction dropout trick, the SAN model shows superior performance in reading comprehension.

Kalchbrenner et al. [19] designed one of the earliest CNN-based methods for sentence modeling, which featured a dynamic CNN (DCNN) that uses the dynamic k-max pooling subsampling and achieves superior performance in sentiment classification. Due to CNN's ability in feature extraction, the DCNN-based system does not require hand-crafted features, which is appreciated and widely adopted by numerous subsequent studies. Kim [4] proposed a simple but effective CNN architecture that utilizes pre-trained word embeddings by word2vec. Kim's work was modified by Liu et al. [20], who propose to learn word embeddings rather than use pre-trained ones directly. Mou et al. designed a tree-based CNN [21] that can capture the general semantics of sentences. In [22], Pang et al. proposed to transform the text matching problem into an image recognition task that can be solved by a CNN-based model. In addition to open-domain datasets, CNNs have also been extensively used in domain-specific tasks, especially in biomedical text classification [22–27].

Chen et al. [18] proposed a two-stage method that combines BiLSTM and CNN for sentiment classification. First, the BiLSTM model is used for sentence type classification. Once assigned a type, a sentence then goes through a 1D CNN layer for sentiment detection. In [28], the authors designed a hybrid network that combines RNN, MLP, and CNN to explore semantic information at each hierarchical level of a document.

2.3. Transformer-Based Models for Social Text Learning

BERT [29] and its variants [30–36] have been extensively used as building blocks for numerous applications, owing to their ability to capture contextual word embeddings. FakeBERT [37] combines BERT and 1D CNN layers to detect fake news in social media. A similar work [38] adopts BERT to detect auto-generated tweets. Mozafari et al. [39] designed a BERT-based transfer learning method to detect hate speech on Twitter. Eke et al. [40] employed BERT to build a sarcasm detector that can classify sarcastic utterances, which is crucial for downstream tasks like sentiment analysis and opinion mining.

2.4. Learning-Based Disaster Tweets Detection

One of the early efforts to identify and classify disaster Tweets is by Stowe et al. [6], who focused on the Tweets generated when Hurricane Sandy hit New York in 2012. In [6], six fine-grained categories of Tweets, including Reporting, Sentiment, Information, Action, Preparation, and Movement, are annotated. With a series of hand-crafted features, such as key terms, Bigrams, time, and URLs, the dataset is used to train three feature-based models, including SVM, maximum entropy, and Naive Bayes models. Palshikar et al. [41] developed a weakly-supervised model based on a bag of words, combined with an online algorithm that helps learn the weights of words to boost detection performance. Algur et al. [42] first transformed Tweets into vectors using count vectorization and Term Frequency-Inverse Document Frequency (TF-IDF), based on a set of pre-identified disaster keywords; the vectorized Tweets are then trained using Naive Bayes, Logistic Regression, J48, Random Forest, and SVM to obtain various classifiers. Singh et al. [43] investigated a Markov model-based model to predict the priority and location of Tweets during a disaster. Madichetty et al. [44] designed a neural architecture that consists of a CNN to extract features from Tweets and a multilayer perceptron (MLP) to perform classification. Joao [45] developed a BERT-based hybrid model that uses both hand-crafted features and learned ones for informative Tweets identification. Li et al. [46] investigate a domain-adapted learning task that uses a Naive Bayes classifier, combined with an iterative self-training algorithm, to incorporate annotated data from a source disaster dataset and data without annotation from the target disaster dataset into a classifier for the target disaster. More broadly, prior efforts on event Tweet detection are also of interest. Ansah et al. [47] proposed a model named SensorTree to detect protest events by tracking information propagated through the Twitter user communities and monitoring the sudden change in the growth of these communities as burst for event detection. Saeed et al. [48] developed a Dynamic Heartbeat Graph (DHG) model to detect trending topics from the Twitter stream. An investigation of recent efforts [49] in disaster Tweet detection reveals a lack of deep learning-based methods that have shown superiority in numerous other NLP applications, as mentioned in Section 2.1. However, in the sub-field of disaster Tweets detection, the use cases are still insufficient. In addition, the idea of integrating sentiment information into a disaster detector remains unexplored, and our study is an attempt to fill this gap.

Inspired by the prior efforts, we design a learning pipeline that includes a BERT variant named SentiBERT [50] to obtain sentiment-aware contextual embeddings, a BiLSTM layer for sequential modeling, and a CNN for feature extraction. The pipeline aggregates the strength of each individual block to enhance the predictive power that realizes an accurate disaster detector.

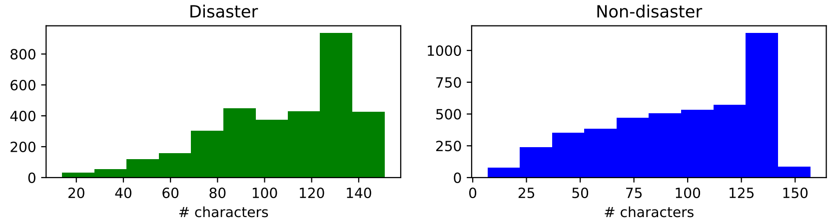
3. Material and Methods

3.1. Dataset

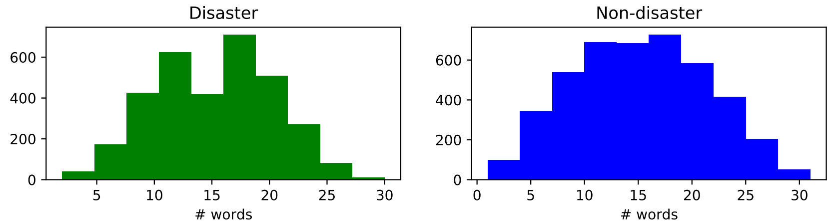
The dataset was created by Figure Eight inc. (an Appen company) from Twitter data and used as a Kaggle competition hosted at https://www.kaggle.com/c/nlp-getting-started/data (accessed on 21 June 2021). There are 10,876 samples in the dataset, including 4692 positive samples (disaster) and 6184 negative samples (not a disaster). Table 1 shows four positive and four negative samples. It can be seen that the disaster and non-disaster Tweets could use similar keywords in different contexts, resulting in different interpretations. For example, "pileup" in sample 1, "airplane's accident" in sample 2, "Horno blaze" in sample 3, and the phrase "a sign of the apocalypse" in sample 4 are more indicative of a disaster. However, the words "bleeding", "blaze", "ambulance", and "Apocalypse" in samples 4 through 8 do not indicate a disaster, given their contexts. Figure 2 displays the histograms of three variables per Tweet: the number of characters, the number of words, and the average number of word lengths. Specifically, the means of the character number per Tweet for disaster and non-disaster Tweets are 108.11 and 95, respectively; the means of the word number per Tweet for disaster and non-disaster Tweets are 15.16 and 14.7, respectively; the means of the average word length for disaster and non-disaster Tweets are 5.92 and 5.14, respectively. The stats data show that the disaster Tweets are relatively longer than the non-disaster ones.

Table 1. Disaster Tweets dataset samples. A + sign indicates a positive sample, and a − sign indicates a negative sample.

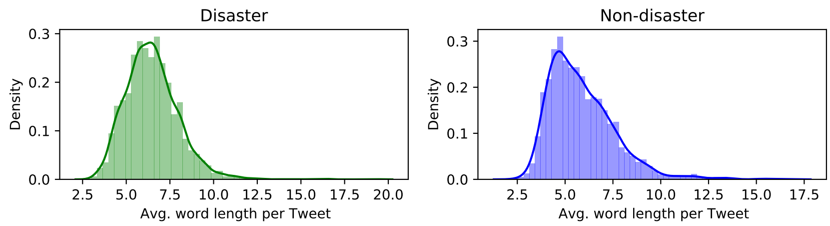
ID	Sample Tweet	Class
1	Grego saw that pileup on TV keep racing even bleeding.	+
2	Family members who killed in an airplane's accident.	+
3	Pendleton media office said only fire on base right now is the Horno blaze.	+
4	I know it's a question of interpretation but this is a sign of the apocalypse.	+
5	bleeding on the brain don't know the cause.	−
6	alrighty Hit me up and we'll blaze!!	−
7	waiting for an ambulance.	−
8	Apocalypse please.	−



(**a**)



(**b**)



(**c**)

Figure 2. Stats of Tweets in the dataset. Histograms of (**a**) the number of characters per Tweet, (**b**) the number of words per Tweet, and (**c**) the average word length per Tweet, plotted for disaster Tweets (**left**) and non-disaster Tweets (**right**).

3.2. Data Pre-Processing

The raw data obtained from Twitter have noises that need to be cleaned. Thus, we apply a pre-processing step to remove the hashtags, emoticons, and punctuation marks. For example, a message "# it's cool. :)", becomes "it's cool." after the filtering. We then apply some basic transformations such as changing "We've" to "We have" to create a better word separation within a sentence. Finally, we tokenize each message to generate a word sequence as the input of the learning pipeline.

3.3. Overview of the Proposed Learning Pipeline

Figure 3 shows the proposed SentiBERT-BiLSTM-CNN learning pipeline, which consists of three sequential modules:

1. SentiBERT is utilized to transform word tokens from the raw Tweet messages to contextual word embeddings. Compared to BERT, SentiBERT is better at understanding and encoding sentiment information.
2. BiLSTM is adopted to capture the order information as well as the long-dependency relation in a word sequence.
3. CNN acts as a feature extractor that strives to mine textual patterns from the embeddings generated by the BiLSTM module.

The output of the CNN is fed to a detection layer to generate the final prediction result, i.e., disaster or not.

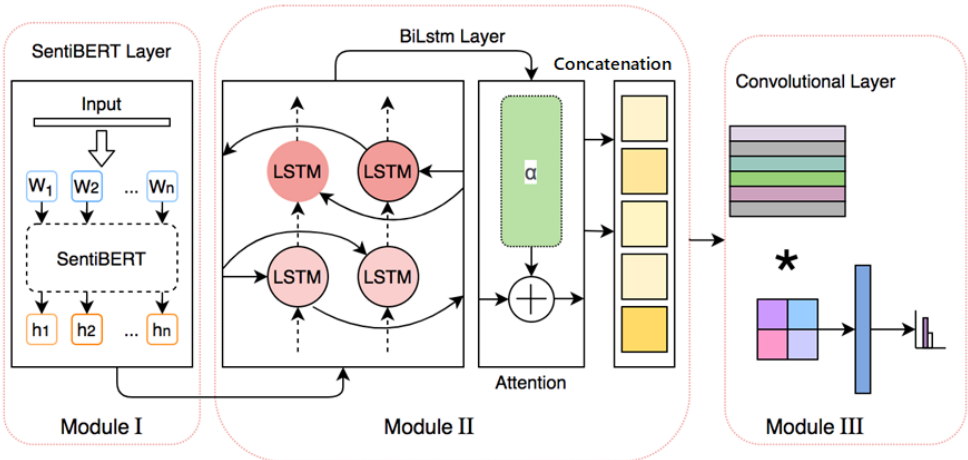


Figure 3. An overview of the SentiBERT-BiLSTM-CNN learning pipeline.

3.4. Sentibert

BERT [29] is an attention-based language model that utilizes a stack of Transformer encoders and decoders to learn textual information. It also uses a multi-headed attention mechanism to extract useful features for the task. The bidirectional Transformer neural network, as the encoder of BERT, converts each word token into a numeric vector to form a word embedding, so that words that are semantically related would be translated to embeddings that are numerically close. BERT also employs a mask language model (MLM) technique and a next sentence prediction (NSP) task in training to capture word-level and sentence-level contextual information. BERT and its variants have been applied to numerous NLP tasks such as named entity recognition, relation extraction, machine translation, and question and answering, and achieved the state-of-the-art performance. In this study, we choose a BERT variant, SentiBERT, which is a transferable transformer-based architecture dedicated to the understanding of sentiment semantics. As shown in Figure 4, SentiBERT modifies BERT by adding a semantic composition unit and a phrase node prediction unit. Specifically, the semantic composition unit aims to obtain phrase representations that are guided by contextual word embeddings and an attentive constituency parsing tree. Phrase-level sentiment labels are used for phrase node prediction. Due to the addition of phrase-level sentiment detection, a sentence can be broken down and analyzed at a finer granularity to capture more sentiment semantics.

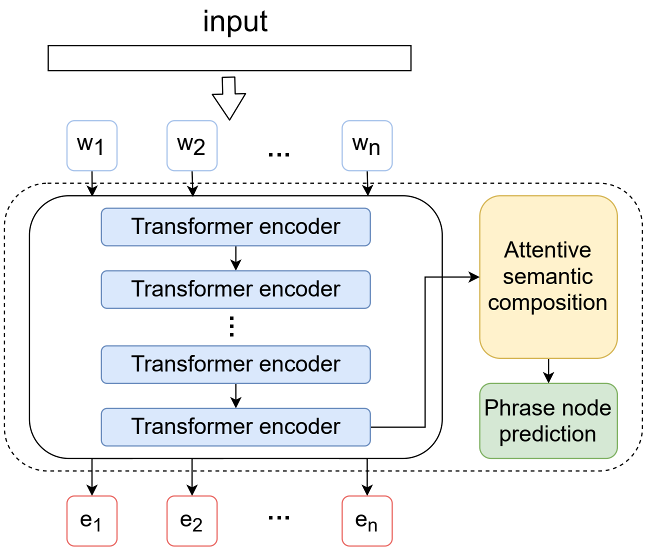


Figure 4. Module I: SentiBERT.

Let $\mathbf{s} = \{w_i | i = 1, ..., n\}$ denote a Tweet message with n word tokens, which are the input of SentiBERT. Our goal is to leverage the power of SentiBERT to generate sentiment-enhanced word embeddings, which can be denoted by $\mathbf{e} = \{e_i = \text{SentiBERT}(w_i) | i = 1, ..., n\}$. In this study, each Tweet should have no more than 64 tokens; the Tweets with less than 64 tokens are padded, namely, $n = 64$. Reference [29] experimentally showed that the output of the last four hidden layers of BERT encodes more contextual information than that of the previous layers. To this end, we also chose a concatenation of the outputs of the last four hidden layers as the word embedding representation.

3.5. Bilstm with Attention

A regular LSTM unit consists of a cell, an input gate, an output gate and a forget gate. The cell can memorize values over arbitrary time periods, and the three gates regulate information flow into and out of the cell to keep what matters and forget what does not. The BiLSTM consists of a forward and a backward LSTM that process an input token vector from both directions. By looking at past and future words, a BiLSTM network can potentially capture the more semantic meaning of a sentence. In our study, the word embeddings $\mathbf{e}$ produced from module I are fed into a standard BiLSTM layer to generate a list of hidden states $\mathbf{h} = \{h_i | i = 1, ..., n\}$, where h_i is given by Equation set (1).

$$\begin{aligned} \overleftarrow{h_i} &= \overleftarrow{\text{LSTM}}(e_i, \overleftarrow{h_{i-1}}) \\ \overrightarrow{h_i} &= \overrightarrow{\text{LSTM}}(e_i, \overrightarrow{h_{i-1}}) \\ h_i &= [\overleftarrow{h_i}; \overrightarrow{h_i}] \end{aligned} \tag{1}$$

where $[;]$ is a concatenation operation. The structure is shown in Figure 5.

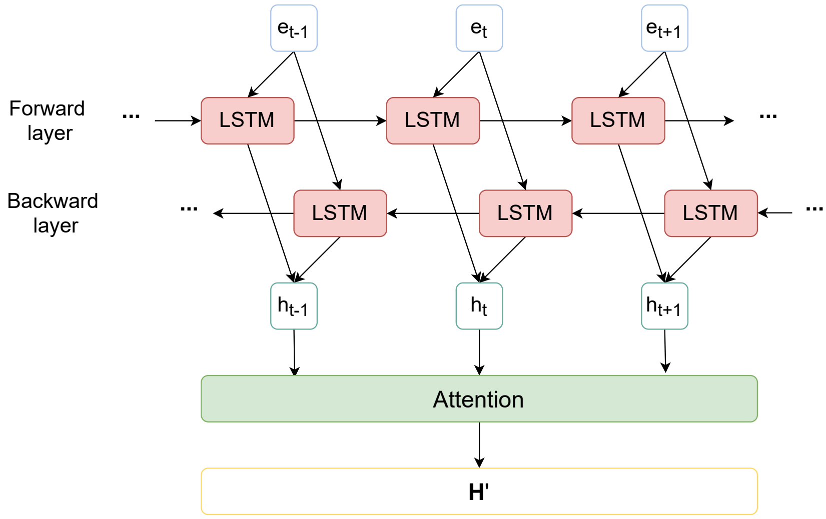


Figure 5. Module II: BiLSTM with attention.

In a Tweet, each word influences the disaster polarity differently. Using an attention mechanism can help the model learn to assign different weights to different words so that the more influential words are given higher weights. For a hidden state h_i, its attention a_i is given in the Equation set (2).

$$\begin{aligned} u_i &= \tanh(W \cdot h_i + b) \\ a_i &= \frac{e^{u_i^\top \cdot u_w}}{\sum_i e^{u_i^\top \cdot u_w}}, \end{aligned} \tag{2}$$

where W denotes a weight matrix, b denotes the bias, and u_w a global context vector, and all three are learned during training. The output of module II is a concatenation of attentive hidden states $\mathbf{H}' = [a_1 h_1; ...; a_n h_n]$.

3.6. CNN

Module III is a CNN that extracts local features, as shown in Figure 6. We adopt a 1D convolutional layer with four differently-sized filters. Each filter scans the input matrix $\mathbf{H}'$ and performs a 1D convolutional along the way to generate a feature map. The extracted features are then fed into a max-pooling layer and concatenated to form a feature matrix $\mathbf{F}$. Lastly, we send a concatenation of $\mathbf{H}'$ and $\mathbf{F}$ to the dense layer.

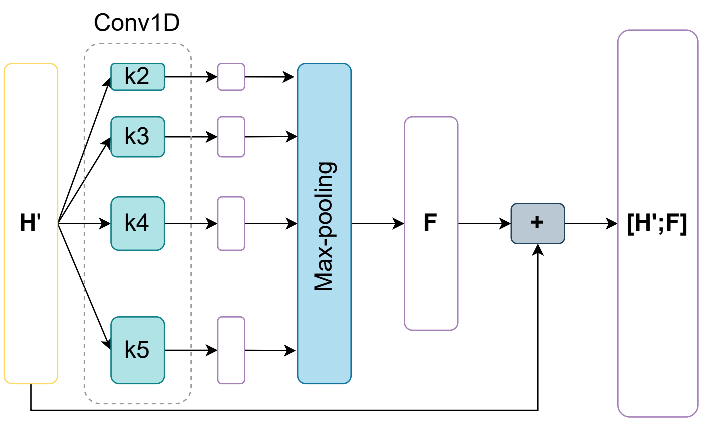


Figure 6. Module III: feature extraction via a CNN layer.

3.7. A Fusion of Loss Functions

In this subsection, we explore the options of loss functions. We considered two individual loss functions including the binary cross-entropy (BCE) loss and the Focal Loss. In addition, we employed a fusion strategy as suggested in [51] to combine the two losses, which resulted in performance improvement.

Since the disaster Tweet detection task is a typical binary classification problem, it is intuitive to utilize the BCE loss as shown in Equation (3) below.

$$L_{BCE} = -\frac{1}{m}\sum_{i=1}^{m}(y_{(i)}log(\hat{y}_{(i)}) + (1 - y_{(i)})log(1 - \hat{y}_{(i)})), \tag{3}$$

in which m is the training set size, and $y_{(i)}$ and $\hat{y}_{(i)}$ denote the ground truth and the predicted class for the ith sample in the dataset, respectively.

Meanwhile, considering the imbalanced sample distribution, this study also employs Focal Loss, defined in Equation (4).

$$L_{FL} = -\frac{1}{m}\sum_{i=1}^{m}(y_{(i)}\alpha(1 - \hat{y}_{(i)})^{\gamma} + (1 - y_{(i)})(1 - \alpha)\hat{y}_{(i)}^{\gamma}log(1 - \hat{y}_{i})), \tag{4}$$

where γ is a coefficient that controls the curve shape of the focal loss function. Using Focal Loss with $\gamma > 1$ reduces the loss for well-classified examples (i.e., with a prediction probability larger than 0.5) and increases loss for hard-to-classify examples (i.e., with a prediction probability less than 0.5). Therefore, it turns the model's attention towards the rare class in case of class imbalance. On the other hand, a lower α value means that we tend to give a small weight to the dominating or common class and high weight to the rare class. By fusing the focal loss and the BCE loss in a certain ratio, we obtain Equation (5), in which β_1 and β_2 specify the fusion weights.

$$L_{mix} = \beta_1 L_{BCE} + \beta_2 L_{FL} \tag{5}$$

4. Experiments

We utilize the disaster Tweet dataset discussed in Section 3.1 for performance evaluation. We first present the performance metrics and then report the experimental results.

4.1. Evaluation Metrics

We use precision (Pre), recall (Rec), and the F1 score to evaluate the model performance. Given that the positive/negative samples are not balanced, F1 is a better metric than accuracy. Precision and recall are also important. The former reflects the number of false alarms; the higher the precision, the fewer false alarms. The latter tells the number of positive samples that are missed; the higher the recall, the fewer disaster Tweets missed. A large precision–recall gap should be avoided, since it indicates that a model focuses on a single metric, while a model should really focus on optimizing F1, the harmonic mean of precision and recall.

Let TP, TN, and FP denote the number of true positives, true negatives, and false positives, respectively, we can then calculate precision, recall, and F1 as follows.

$$Pre = \frac{TP}{TP + FP} \tag{6}$$

$$Rec = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = 2 \times \frac{Pre \times Rec}{Pre + Rec}. \tag{8}$$

4.2. Training Setting

The dataset was divided into training and validation sets in the ratio of 7:3, generating 7613 training and 3263 validation samples. For the SentiBERT, the embedding dimension was 768, max sequence length was 128, and layer number was 12; for the BiLSTM module, the layer number was 1, and the feature number was 768; for the CNN module, the sizes of the four filters were set to 2, 3, 4 and 5. For the overall architecture, we used a learning rate of 1×10^{-4}, the Adam optimizer, and experimented with different batch sizes (16 and 32) and training epochs (6, 8, 10, 12, and 14). All experiments were implemented using Python 3.9.4 and PyTorch 1.8.0 on Google Colab with an NVIDIA Tesla K80.

4.3. Baseline Model

The baseline model we chose was a BERT-based hybrid model developed by Joao [45]. We denote the model as BERT_{hyb}. We regard BERT_{hyb} as a credible baseline because it presented the state-of-the-art (SOTA) performance compared to a variety of models on four datasets. BERT_{hyb} works by combining a series of hand-crafted Tweet features and the BERT word embeddings and sending the feature concatenation to an MLP for classification.

4.4. Effect of Hyper-Parameter Choices

We conducted experiments to evaluate the performance of our model SentiBERT-BiLSTM-CNN under different hyper-parameter settings. Specifically, the model was trained with a combination of three values of epochs (6, 8, 10, 12, and 14) and two values of batch sizes (16, 32), creating ten experiments, as shown in Table 2. It can be seen that when the model was trained with 10 epochs and with a batch size of 32, the model achieved the best performance, with an F1 of 0.8956. We also observe a consistent performance improvement as the number of epochs increases from 6 to 10, and beyond 10 epochs, the gain is not apparent. The training was efficient because SentiBERT has been pre-trained and was only fine-tuned on our dataset. It is noted that for this set of experiments, we applied a basic cross-entropy loss function. The effect of the fused loss function is reported in the next subsection.

Table 2. Performance of SentiBERT-BiLSTM-CNN under different hyper-parameter settings.

Epochs	Batch Size	Precision	Recall	F1 Score
6	32	0.8525	0.8478	0.8501
	16	0.8495	0.8364	0.8429
8	32	0.8654	0.8701	0.8677
	16	0.8643	0.8618	0.8630
10	32	**0.8987**	0.8932	**0.8959**
	16	0.8903	0.8827	0.8865
12	32	0.8848	0.8956	0.8902
	16	0.8817	0.8893	0.8855
14	32	0.8902	**0.9012**	0.8957
	16	0.8949	0.8878	0.8913

4.5. The Effect of a Hybrid Loss Function

We conducted experiments to evaluate the model's performance under different loss function settings. We first evaluated the performance of using BCE and FL individually and then fused the two loss functions in the ratio of 1:1. The results are reported in Table 3. We observe that the model with FL outperformed the model with BCE, validating the efficacy of FL in the case of imbalanced data distribution. In addition, the model with a hybrid loss function performed the best, with an F1 of 0.9275. The result demonstrates the effectiveness of the fusion strategy in this study.

Table 3. Performance of SentiBERT-BiLSTM-CNN under different loss function settings.

Loss Function	Epochs	Batch Size	Precision	Recall	F1 Score
L_{BCE}	10	32	0.8987	0.8932	0.8959
L_{FL}	10	32	0.9029	0.9135	0.9082
L_{mix}	10	32	**0.9305**	**0.9271**	**0.9275**

4.6. Performance Evaluation

We also conducted experiments to evaluate a set of models, and present a performance comparison of all evaluated models in Table 4, using the best hyper-parameter settings and the fused loss function, as reported in the previous two subsections. We give the result analysis as follows.

- The set of models CNN, BiLSTM, SentiBERT, BiLSTM-CNN, and SentiBERT-BiLSTM-CNN forms an ablation study, from which we can evaluate the performance of each individual module and the combined versions. It can be seen that the pure CNN model performs the worst since a single-layer CNN cannot learn any contextual information. Both BiLSTM (with attention) and SentiBERT present an obvious improvement. SentiBERT is on a par with BiLSTM-CNN in precision, but outperforms it in recall. Our final model, SentiBERT-BiLSTM-CNN tops every other model, showing its power to combine the strength of each individual building block.
- The set of models fastText-BiLSTM-CNN, word2vec-BiLSTM-CNN, BERT-BiLSTM-CNN, and SentiBERT-BiLSTM-CNN are evaluated to compare the effect of word embeddings. FastText [52], word2vec [53], BERT, and SentiBERT are used for the same purpose, i.e., to generate word embeddings. A model's ability to preserve contextual information determines its performance. From the results, we observe that by adding contextual embeddings, the models gain improvements to varying degrees. SentiBERT-BiLSTM-CNN, as the best-performing model, demonstrates superior capability in encoding contextual information.
- Another observation is that SentiBERT-BiLSTM-CNN outperforms BERT-BiLSTM-CNN by 1.23% in F1, meaning that sentiment in Tweets is a crucial factor that can help detect disaster Tweets, and a sentiment-enhanced BERT validates this hypothesis.
- Lastly, SentiBERT-BiLSTM-CNN outperforms BERT_{hyb}, i.e., the SOTA, by 0.77% in F1. Although BERT_{hyb} presented the highest precision 0.9413, its precision–recall gap (4.21%) is large, compared to that of SentiBERT-BiLSTM-CNN (0.34%), meaning that BERT_{hyb} focuses more on optimizing precision. On the other hand, SentiBERT-BiLSTM-CNN demonstrated a more balanced result in precision and recall.

Table 4. A performance comparison of models.

Model	Precision	Recall	F1 Score
CNN	0.8064	0.8086	0.8025
BiLSTM	0.8571	0.8405	0.8487
SentiBERT	0.8668	0.8712	0.8690
BiLSTM-CNN	0.8674	0.8523	0.8598
word2vec-BiLSTM-CNN	0.8831	0.8767	0.8799
fastText-BiLSTM-CNN	0.8935	0.8736	0.8834
BERT-BiLSTM-CNN	0.9118	0.9187	0.9152
BERT_{hyb}	**0.9413**	0.8992	0.9198
SentiBERT-BiLSTM-CNN	0.9305	**0.9271**	**0.9275**

4.7. Error Analysis

Table 5 shows ten samples, including five positive and five negative ones, which are misclassified by the proposed SentiBERT-BiLSTM-CNN model. In this subsection, we provide an analysis of these mistakes that may shed light on further improvement of our model.

- For the five samples that are marked as disaster Tweets (i.e., samples one through five), none of them are describing a common sense disaster: sample 1 seems to state a personal accident; sample 2 talks about US dollar crisis which may indicate inflation given its context; in sample 3, the phrase "batting collapse" refers to a significant failure of the batting team in a sports game; sample 4 is the closest to a real disaster, but the word "simulate" simply reverses the semantic meaning; sample 5 does mention a disaster "Catastrophic Man-Made Global Warming", but the user simply expresses his/her opinion against it. Our observation is that the process of manual annotation could introduce some noises that would affect the modeling training. From another perspective, the noises help build more robust classifiers and potentially reduce overfitting.
- For the five negative samples (6–10), we also observe possible cases of mislabeled samples: sample 6 clearly reports a fire accident with the phrase "burning buildings" but was not labeled as a disaster Tweet; sample 7 states a serious traffic accident; sample 8 mentions bio-disaster with the phrase "infectious diseases and bioterrorism"; sample 9 has only three words, and it is hard to tell its class without more context, although the word "bombed" is in the Tweet; sample 10 reflects a person's suicide intent, which could have been marked as a positive case.

Table 5. Examples of misclassified samples. A "+" sign indicates a positive sample, and a "−" sign indicates a negative sample.

ID	Sample Tweet	Label	Prediction
1	I was wrong to call it trusty actually. considering it spontaneously collapsed on me that's not very trusty.	+	−
2	Prices here are insane. Our dollar has collapsed against the US and it's punishing us. Thanks for the info.	+	−
3	Now that's what you call a batting collapse.	+	−
4	Emergency units simulate a chemical explosion at NU.	+	−
5	99% of Scientists don't believe in Catastrophic Man-Made Global Warming only the deluded do.	+	−
6	all illuminated by the brightly burning buildings all around the town!	−	+
7	That or they might be killed in an airplane accident in the night a car wreck! Politics at it's best.	−	+
8	automation in the fight against infectious diseases and bioterrorism	−	+
9	misfit got bombed.	−	+
10	Because I need to know if I'm supposed to throw myself off a bridge for a #Collapse or plan the parade. There is no both.	−	+

We need to clarify that these misclassified samples presented in the table are randomly selected from all error predictions. It can be seen that the length limit of Tweets presents pros and cons for training a classifier. The bright side is that users are forced to use short and direct words to express an opinion, and the downside is that some short Tweets are hard to interpret due to the lack of more context information, which is the main challenge for training an accurate model.

5. Conclusions

Disaster analysis is highly related to people's daily lives, and recent years have seen more research efforts dedicating to this field. Research on disaster prediction helps augment people's awareness, improve the mechanism of a government rescue, and schedule charitable institutions' work. This paper investigates a novel model for disaster detection using Tweets. Our model, SentiBERT-BiLSTM-CNN, leverages a sentiment-aware BERT encoder, an attentive BiLSTM, and a 1D convolutional layer to extract high-quality linguistic features for disaster prediction. The model is validated through extensive experiments compared to its peers, making it a competitive model for building a real-time disaster detector.

Although the proposed model is trained and validated on an English dataset, it can be applied to datasets in other languages. Specifically, in a different language environment, the following adjustments need to be made: first, we should find a BERT model pre-trained in the target language or in a multi-lingual setting, which is readily available online (https://huggingface.co/transformers/pretrained_models.html, accessed on 12 March 2021); second, we need to retrain SentiBERT on a sentiment analysis dataset in the target language; lastly, a new disaster Tweet dataset in the target language is needed to train and validate the model. In this new language environment, SentiBERT can now generate sentiment-aware word embeddings to be consumed by the subsequent BiLSTM and CNN modules, which are language independent.

This work has the following limitations that also point out the future directions. First, it remains interesting to uncover the role keywords played in disaster detection. Given that keywords like "blaze" and "apocalypse" can appear in both disaster and non-disaster Tweets, it is challenging to effectively utilize the keywords as extra knowledge to help boost the detection accuracy. One potential solution is to fine-tune BERT through pair-wise training, taking a pair of Tweets containing the same keywords but with opposite training labels; this way, BERT is forced to better understand the contextual difference between two Tweets. Second, it remains unknown that how well the model trained on our dataset performs on other disaster datasets, such as HumAID [54] and Crisismmd [55]; in addition, we expect to obtain a more robust model that is trained across multiple disaster/crisis Tweets datasets. Third, we are interested in creating a multilingual disaster detector that can understand and process Tweets in different languages; it is worth conducting a performance comparison between a multilingual and a monolingual model.

Author Contributions: Conceptualization and methodology, G.S. and D.H.; software, validation, and original draft preparation, G.S.; review and editing, supervision, funding acquisition, D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key Research and Development Program of China (2020AAA0108004) and the National Natural Science Foundation of China (61672127, U1936109). The funding agency has no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data Availability Statement: Natural Language Processing with Disaster Tweets dataset supporting the conclusions of this article are available at https://www.kaggle.com/c/nlp-getting-started/data (accessed on 20 March 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, C.; Weng, J.; He, Q.; Yao, Y.; Datta, A.; Sun, A.; Lee, B.S. Twiner: Named entity recognition in targeted twitter stream. In Proceedings of the 35th international ACM SIGIR Conference on Research and Development in Information Retrieval, Portland, OR, USA, 12–16 August 2012; pp. 721–730.
2. Ritter, A.; Wright, E.; Casey, W.; Mitchell, T. Weakly supervised extraction of computer security events from twitter. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 896–905.

3. Soulier, L.; Tamine, L.; Nguyen, G.H. Answering twitter questions: A model for recommending answerers through social collaboration. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 267–276.
4. Kim, Y. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1746–1751.
5. Steinskog, A.; Therkelsen, J.; Gambäck, B. Twitter topic modeling by tweet aggregation. In Proceedings of the 21st Nordic Conference on Computational Linguistics, Gothenburg, Sweden, 22–24 May 2017; pp. 77–86.
6. Stowe, K.; Paul, M.; Palmer, M.; Palen, L.; Anderson, K.M. November. Identifying and categorizing disaster-related tweets. In Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media, Austin, TX, USA, 1 November 2016; pp. 1–6.
7. Bakshi, R.K.; Kaur, N.; Kaur, R.; Kaur, G. Opinion mining and sentiment analysis. In Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 March 2016; pp. 452–455.
8. Go, A.; Bhayani, R.; Huang, L. Twitter sentiment classification using distant supervision. CS224N project report. *Stanford* **2009**, *1*, 2009.
9. Kouloumpis, E.; Wilson, T.; Moore, J. Twitter sentiment analysis: The good the bad and the omg! In Proceedings of the International AAAI Conference on Web and Social Media, Catalonia, Spain, 17–21 July 2011; Volume 5.
10. Hao, Y.; Mu, T.; Hong, R.; Wang, M.; Liu, X.; Goulermas, J.Y. Cross-domain sentiment encoding through stochastic word embedding. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 1909–1922. [CrossRef]
11. Sankaranarayanan, J.; Samet, H.; Teitler, B.E.; Lieberman, M.D.; Sperling, J. Twitterstand: News in tweets. In Proceedings of the 17th ACM Sigspatial International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 4–6 November 2009; pp. 42–51.
12. Sriram, B.; Fuhry, D.; Demir, E.; Ferhatosmanoglu, H.; Demirbas, M. Short text classification in twitter to improve information filtering. In Proceedings of the 33rd international ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 19–23 July 2010; pp. 841–842.
13. Yin, J.; Lampert, A.; Cameron, M.; Robinson, B.; Power, R. Using social media to enhance emergency situation awareness. *IEEE Ann. Hist. Comput.* **2012**, *27*, 52–59. [CrossRef]
14. Kogan, M.; Palen, L.; Anderson, K.M. Think local, retweet global: Retweeting by the geographically-vulnerable during Hurricane Sandy. In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing, Vancouver, BC, Canada, 14–18 March 2015; pp. 981–993.
15. Lamb, A.; Paul, M.; Dredze, M. Separating fact from fear: Tracking flu infections on twitter. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; pp. 789–795.
16. Verma, S.; Vieweg, S.; Corvey, W.; Palen, L.; Martin, J.; Palmer, M.; Schram, A.; Anderson, K. Natural language processing to the rescue? extracting "situational awareness" tweets during mass emergency. In Proceedings of the International AAAI Conference on Web and Social Media, Catalonia, Spain, 17–21 July 2011; Volume 5.
17. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
18. Chen, T.; Xu, R.; He, Y.; Wang, X. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Syst. Appl.* **2017**, *72*, 221–230. [CrossRef]
19. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. *arXiv* **2014**, arXiv:1404.2188.
20. Liu, J.; Chang, W.C.; Wu, Y.; Yang, Y. Deep learning for extreme multi-label text classification. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 115–124.
21. Mou, L.; Men, R.; Li, G.; Xu, Y.; Zhang, L.; Yan, R.; Jin, Z. Natural language inference by tree-based convolution and heuristic matching. *arXiv* **2015**, arXiv:1512.08422.
22. Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Wan, S.; Cheng, X. Text matching as image recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
23. Wang, J.; Wang, Z.; Zhang, D.; Yan, J. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. In Proceedings of the IJCAI, Melbourne, Australia, 19–25 August 2017; Volume 350.
24. Karimi, S.; Dai, X.; Hassanzadeh, H.; Nguyen, A. Automatic diagnosis coding of radiology reports: A comparison of deep learning and conventional classification methods. In Proceedings of the BioNLP 2017, Vancouver, BC, Canada, 4 August 2017; pp. 328–332.
25. Peng, S.; You, R.; Wang, H.; Zhai, C.; Mamitsuka, H.; Zhu, S. DeepMeSH: Deep semantic representation for improving large-scale MeSH indexing. *Bioinformatics* **2016**, *32*, i70–i79. [CrossRef] [PubMed]
26. Rios, A.; Kavuluru, R. Convolutional neural networks for biomedical text classification: application in indexing biomedical articles. In Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics, Atlanta, GA, USA, 9–12 September 2015; pp. 258–267.
27. Hughes, M.; Li, I.; Kotoulas, S.; Suzumura, T. Medical text classification using convolutional neural networks. *Stud. Health Technol. Inform.* **2017**, *235*, 246–250. [PubMed]

28. Kowsari, K.; Brown, D.E.; Heidarysafa, M.; Meim, i K.J.; Gerber, M.S.; Barnes, L.E. Hdltex: Hierarchical deep learning for text classification. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 364–371.
29. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
30. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
31. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
32. Joshi, M.; Chen, D.; Liu, Y.; Weld, D.S.; Zettlemoyer, L.; Levy, O. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 64–77. [CrossRef]
33. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
34. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Tian, H.; Wu, H.; Wang, H. Ernie 2.0: A continual pre-training framework for language understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8968–8975.
35. Liu, X.; Cheng, H.; He, P.; Chen, W.; Wang, Y.; Poon, H.; Gao, J. Adversarial training for large neural language models. *arXiv* **2020**, arXiv:2004.08994.
36. Graves, A.; Jaitly, N.; Mohamed, A.R. Hybrid speech recognition with deep bidirectional LSTM. In Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–12 December 2013; pp. 273–278.
37. Kaliyar, R.K.; Goswami, A.; Narang, P. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimed. Tools Appl.* **2021**, *80*, 11765–11788. [CrossRef]
38. Harrag, F.; Debbah, M.; Darwish, K.; Abdelali, A. Bert transformer model for detecting Arabic GPT2 auto-generated tweets. *arXiv* **2021**, arXiv:2101.09345.
39. Mozafari, M.; Farahbakhsh, R.; Crespi, N. A BERT-based transfer learning approach for hate speech detection in online social media. In *International Conference on Complex Networks and Their Applications*; Springer: Cham, Switzerland, 2019; pp. 928–940.
40. Eke, C.I.; Norman, A.A.; Shuib, L. Context-Based Feature Technique for Sarcasm Identification in Benchmark Datasets Using Deep Learning and BERT Model. *IEEE Access* **2021**, *9*, 48501–48518. [CrossRef]
41. Palshikar, G.K.; Apte, M.; P.; ita, D. Weakly supervised and online learning of word models for classification to detect disaster reporting tweets. *Inf. Syst. Front.* **2018**, *20*, 949–959. [CrossRef]
42. Algur, S.P.; Venugopal, S. Classification of Disaster Specific Tweets-A Hybrid Approach. In Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 17–19 March 2021; pp. 774–777.
43. Singh, J.P.; Dwivedi, Y.K.; Rana, N.P.; Kumar, A.; Kapoor, K.K. Event classification and location prediction from tweets during disasters. *Ann. Oper. Res.* **2019**, *283*, 737–757. [CrossRef]
44. Madichetty, S.; Sridevi, M. Detecting informative tweets during disaster using deep neural networks. In Proceedings of the 2019 11th International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, India, 7–11 January 2019; pp. 709–713.
45. Joao, R.S. On Informative Tweet Identification For Tracking Mass Events. *arXiv* **2021**, arXiv:2101.05656.
46. Li, H.; Caragea, D.; Caragea, C.; Herndon, N. Disaster response aided by tweet classification with a domain adaptation approach. *J. Contingencies Crisis Manag.* **2018**, *26*, 16–27. [CrossRef]
47. Ansah, J.; Liu, L.; Kang, W.; Liu, J.; Li, J. Leveraging burst in twitter network communities for event detection. *World Wide Web* **2020**, *23*, 2851–2876. [CrossRef]
48. Saeed, Z.; Abbasi, R.A.; Razzak, I. Evesense: What can you sense from twitter? In *European Conference on Information Retrieval*; Springer: Cham, Switzerland, 2020; pp. 491–495.
49. Sani, A.M.; Moeini, A. Real-time Event Detection in Twitter: A Case Study. In Proceedings of the 2020 6th International Conference on Web Research (ICWR), Tehran, Iran, 22–23 April 2020; pp. 48–51.
50. Yin, D.; Meng, T.; Chang, K.W. Sentibert: A transferable transformer-based architecture for compositional sentiment semantics. *arXiv* **2020**, arXiv:2005.04114.
51. Song, G.; Huang, D.; Xiao, Z. A Study of Multilingual Toxic Text Detection Approaches under Imbalanced Sample Distribution. *Information* **2021**, *12*, 205. [CrossRef]
52. Bojanowski, P.; Edouard, G.; Arm, J.; Tomas, M. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
53. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
54. Alam, F.; Qazi, U.; Imran, M.; Ofli, F. HumAID: Human-Annotated Disaster Incidents Data from Twitter with Deep Learning Benchmarks. *arXiv* **2021**, arXiv:2104.03090.
55. Alam, F.; Ofli, F.; Imran, M. Crisismmd: Multimodal twitter datasets from natural disasters. In Proceedings of the International AAAI Conference on Web and Social Media, Stanford, CA, USA, 25–28 June 2018; Volume 12.