Computers & Geosciences 178 (2023) 105405

Contents lists available at ScienceDirect

Computers and Geosciences

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

Real-time social media sentiment analysis for rapid impact assessment of floods

Lydia Bryan-Smith [*], Jake Godsall, Franky George, Kelly Egode, Nina Dethlefs, Dan Parsons

University of Hull, Cottingham Road, Hull, HU6 7RX, United Kingdom

ARTICLE INFO

Keywords:
Social media
Sentiment analysis
Flooding
Artificial Intelligence

ABSTRACT

Traditional approaches to flood modelling mostly rely on hydrodynamic physical simulations. While these simulations can be accurate, they are computationally expensive and prohibitively so when thinking about real-time prediction based on dynamic environmental conditions.

Alternatively, social media platforms such as Twitter are often used by people to communicate during a flooding event, but discovering which tweets hold useful information is the key challenge in extracting information from posts in real time.

In this article, we present a novel model for flood forecasting and monitoring that makes use of a transformer network that assesses the severity of a flooding situation based on sentiment analysis of the multimodal inputs (text and images). We also present an experimental comparison of a range of state-of-the-art deep learning methods for image processing and natural language processing. Finally, we demonstrate that information induced from tweets can be used effectively to visualise fine-grained geographical flood-related information dynamically and in real-time.

1. Introduction

Natural disasters, such as floods, can occur suddenly and without much warning, forcing people to leave their homes, damaging infrastructure, destroying livelihoods, and having long-term impacts on the health of those affected (FitzGerald et al., 2019; Gould et al., 2020; Khayyam and Noureen, 2020). With increasing occurrence of such events due to climate change and associated phenomena, it is imperative to be able to predict floods in an accurate and timely manner — to give early warnings and avert humanitarian disasters, but also to direct help effectively when a flood has occurred.

Traditional approaches rely on hydrodynamic physical simulations. While these simulations can be accurate (Price et al., 2012; Vichiantong et al., 2019), they are computationally expensive and not suited to real-time prediction based on dynamic environmental conditions (Coulthard et al., 2013; Teng et al., 2017).

Social media platforms like Twitter are often used by people to communicate during a flooding event (Kongthon et al., 2012) and other natural disasters (Sakaki et al., 2010; Riddell and Fenner, 2021). While extracting information from such social media posts in real time has the potential to increase situational awareness during flooding events, the key challenge in achieving this however is discovering which tweets hold useful information (e.g. "Part of London Road in Carlisle is closed after a building was badly damaged by #StormFranklin") and which ones do not (e.g. "Never too early for lunch") (Gao et al., 2011).

We present a novel model for flood forecasting and monitoring that can simultaneously process and interpret information from text and images to (a) assess the severity of a flooding situation based on sentiment analysis of the multimodal inputs, and (b) map the development of floods dynamically using geolocations of tweets, in combination with the sentiment analysis computed. We present an experimental comparison of a range of state-of-the-art deep learning methods for image processing and time-series modelling, showing that models that combine text and images achieve superior performance to unimodal models (e.g. text-only or images-only) and information induced from tweets can be used effectively to visualise fine-grained geographical flood-related information dynamically and in real-time.

We make the following key contributions in this article:

- A novel model that uses joint linguistic and visual feature embeddings to create a multimodal representation of sentiment in flood-related tweets on social media.
- We show that geographical and sentiment information induced from tweets can model the dynamics and severity of floods in different geographical areas.

[*] Corresponding author.
E-mail address: L.Bryan-Smith@hull.ac.uk (L. Bryan-Smith).

https://doi.org/10.1016/j.cageo.2023.105405
Received 19 October 2022; Received in revised form 16 June 2023; Accepted 17 June 2023
Available online 28 June 2023

- A set of benchmarks using state-of-the-art deep learning methodology. All code and data (where we are able to share) are publicly available.

2. Related works

2.1. Flood forecasting

Traditionally, flood forecasting has been approached with physics-based models such as LISFLOOD-FP (Coulthard et al., 2013), DLEFT3D (Deltares, 2021), ANUGA (Davies and Roberts, 2015), and others (Ming et al., 2020; Roux et al., 2020; Wu et al., 2020). Taking in current environmental information, these models run a simulation to calculate a future forecast of environmental conditions. These calculations tend to be computationally expensive, taking many hours to complete, and must also be manually calibrated — which can make them prohibitively expensive for real-time use cases.

GeoAI (Remote sensing and AI) (Janowicz et al., 2020; Li, 2020) can alleviate some of these concerns (Keung et al., 2018; Furquim et al., 2018). Multiple approaches have been applied here, for example predicting future sensor values (Le et al., 2019) with an Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), estimating risk with a Support Vector Machine (SVM) (Cortes and Vapnik, 1995), or breaking rivers up into a graph of smaller models predicting environmental conditions in the future (Moshe et al., 2020). These suffer from a number of issues however, from being limited to specific geographical positions (Le et al., 2019; Moshe et al., 2020), relying on computationally costly simulations (Mojaddadi et al., 2017), or limited generalisability (Le et al., 2019) to new and previously unseen locations.

While physics-based models can predict water levels in specific areas, they are not fast enough to forecast and monitor floods dynamically in-situ, e.g. for locations with no historical data available. Also, common to all approaches reviewed is that they do not take humanitarian needs into account, as these are difficult to infer from water depth alone.

2.2. Sentiment analysis from social media

In this article, we aim to explore the possibility of using sentiment analysis to assess flood severity and humanitarian needs in different locations. The idea of applying sentiment analysis to social media data is well established. Lexicon-based approaches are well explored (Baccianella et al., 2010; Mohammad and Turney, 2013; Vashishtha and Susan, 2019; Rout et al., 2018; Hutto and Gilbert, 2014), but are limited to hard preset rules. Other linguistically-inspired approaches opt for a stronger grammatical representation of the input. For example, Fu et al. (2016) use rhetorical structure theory and an LSTM to parse the tweet text — preserving contextual information in a tree-like form. Some projects have attempted fine-grained classification (e.g. into 6 emotional classes “happy”, “sad”, “anger”, “fear”, “surprise”, and “disgust”) (Purver and Battersby, 2012; Schoene and Dethlefs, 2016), but challenges remain with respect to accuracy, which is lower than other binary approaches.

Emojis are also often used to express emotions (Rout et al., 2018), but many existing models fail to take them into account (Sahni et al., 2017; Rout et al., 2018; Kokab et al., 2022). Scope exists to make use of them in sentiment analysis tasks. For example, Felbo et al. (2017) demonstrates a potential approach to address this by deriving positive/negative sentiment labels from emojis in tweets, enabling an LSTM-based model to be trained on a very large dataset of Twitter posts (1.2B tweets). This approach removes the need for manual and keyword-driven annotation, which reduces manual labour requirements and improves representation of the target domain in the training dataset as positive words (e.g. “excited” or “sad”) are not being used as labels.

While LSTMs are powerful at modelling natural language (Felbo et al., 2017; Fu et al., 2016), they are not well suited to being parallelised on a GPU or other parallel computing device, increasing training times and underutilising equipment (Hochreiter and Schmidhuber, 1997; Vaswani et al., 2017). Transformers do not have this limitation (Vaswani et al., 2017) so scope exists to apply them to the problem of sentiment analysis (Agüero-Torales et al., 2021). Zhang et al. (2020) applies Bidirectional Encoder Representations from Transformers (BERT; a transformer-based sentence embedding model, see Section 4.1) (Devlin et al., 2019), Robustly optimised BERT approach (RoBERTa) (Liu et al., 2019), and XLNet (Yang et al., 2019) to both manually, automatically (emojis), and crowdsourced (reviews with author-annotated labels) labelled data to gain performance improvements over baselines, but these models have a large number of parameters (e.g. 110M for BERT (Devlin et al., 2019)), making them memory and computationally expensive.

In comparison to sentiment analysis from other modalities, e.g. news or reviews, social media data faces a number of challenges. These include non-standard spellings (due to typos or abbreviations), non-standard use of words and grammar, rapidly evolving vocabulary, mixed languages and images, urls, usernames, hashtags, etc. Kokab et al. (2022) tries to solve these challenges by splitting words up with BERT as a word embedding to incorporating out-of-vocabulary words, training an LSTM model to predict positive/negative sentiment, but strips punctuation and stop words, potentially losing some semantic meaning. To address this, we propose a transformer-based binary sentiment analysis model using Global Vectors for Word Representation (GloVe) (Pennington et al., 2014) pretrained on multilingual twitter data. In doing so, multilingual text, non-standard grammar, and hashtags are included in sentiment predictions. Further, we use contrastive learning (Radford et al., 2021) to predict sentiment using both text and images at the same time.

2.3. Social sensing for emergencies

Social sensing approaches such as sentiment analysis/monitoring from social media have been applied to emergency situations previously. The problem task can be divided into two components:

1. Identifying new emergency events from social media content, and
2. Providing real-time intelligence on humanitarian needs during a known emergency event, again from information provided online.

Different pieces of related research have focused on either of these problems. For example, Arthur et al. (2018) manually labels a dataset of 3879 tweets to train a Naïve Bayes filter to classify tweets by relevance. Relevant tweets are then geocoded using a number of heuristics, before finally performing burst detection to identify flooding events and plotting geocoded tweets on a map. The small dataset limits the transferability of the relevancy classifier (Li et al., 2017), however. Similarly, Smith et al. (2017) streams tweets and filters them using a keyword (relevancy) and geocoding based approach before then detecting bursts of filtered tweets with a simple threshold. When a burst is detected, a hydrodynamic model is run a period of 4 h with LiDAR and real-time rainfall data as inputs to predict areas which are likely flooded.

In this article, we will focus on part of the second problem task, specifically determining locations of humanitarian needs during an emergency situation by quantifying tweets by their sentiment and location, in our case, a flood.

An approach focusing on the identification of humanitarian needs is taken by Kankanamge et al. (2020), who use frequency analysis and word clustering to discover useful information from tweets during a disaster. This proved effective for their target domain and data, but may

Table 1
Overview of the floods included in our dataset.

	Start date	End date	Region	Tweets
Finsbury park flood	2019-10-08	2021-08-07	London, UK	267
Storm Dennis	2019-12-08	2021-07-22	UK	120,861
Storm Christoph	2021-01-17	2021-07-08	UK	17,085
Storm Jorge	2020-02-27	2021-07-10	UK	25,102
Hurricane Eta	2020-10-30	2021-07-16	Central America	17,265
Hurricane Beta	2020-09-18	2020-11-12	Central America	315
Hurricane Iota	2020-11-09	2022-07-22	Central America	315
New South Wales Floods	2021-02-04	2021-07-23	Australia	15,759
Queensland Floods	2018-12-18	2021-06-28	Australia	1889
Colorado Flooding	2013-08-10	2021-07-23	USA	1420
Mexico Floods	2020-11-07	2021-07-22	Mexico	26
Snaith Floods	2020-02-25	2021-07-13	Snaith, UK	322
Storm Franklin	2022-02-16	2022-03-04	UK	13,851
Himachalpradesh	2021-07-06	2021-07-31	Himachal Pradesh, North India	551
Texas Floods	2021-05-01	2021-05-31	Texas, USA	892
Sydney Floods	2022-02-23	2022-03-21	Sydney, Australia	2553
(floods OR flashfloods)	2007-12-06	2021-07-23	Worldwide	553,218

not be easily generalisable to other events as tweets were manually-labelled (Li et al., 2017). The difficulty in transferring concepts learnt from a disaster in one place to a disaster in another is highlighted by Li et al. (2017). By using transfer learning performance improvements were made in generalising a model to be effective in multiple disasters. This transfer learning approach however assumes that all the tweets from the target disaster will be available up-front, which may not always be the case. Additionally, lower accuracy is observed when transferring between disasters of different types, and only a small dataset (7000–9000 tweets) is used.

Ragini et al. (2018) also classifies tweets using a dictionary: firstly, classifying them by objectivity, secondly categorising by humanitarian need (e.g. water, food, medical emergency, etc.), and finally sentiment analysing the subjective tweets with an SVM. The tweets per category evaluated however is again small (2000 for the majority class) and with unbalanced categories the performance (F1: 0.95) is not directly comparable to other studies.

Avvenuti et al. (2014) instead filters tweets by relevancy (i.e. "useful" and "not useful") using a decision tree pretrained on a static dataset, before then performing burst detection to detect events and extracting and geocoding place names to determine where the event happened. The decision tree used though is trained on 1412 manually labelled tweets, which as Li et al. (2017) suggests limits the generalisability of the approach.

Alternative approaches used include clustering and visualisation (Beigi et al., 2016), burst detection (Yin et al., 2012) analysing images (Ning et al., 2020; de Vitry et al., 2019), hand-labelling training datasets (Avvenuti et al., 2014), or use keyword-based analysis (Arthur et al., 2018; Ragini et al., 2018; Smith et al., 2017), which does not generalise easily to new and sudden events (Li et al., 2017), but despite this wide range of approaches being taken to the issue, limited attempts to combine text and images have been taken (Wang et al., 2018; Said et al., 2020).

We suggest that scope exists to apply modern machine learning algorithms such as transformers (Vaswani et al., 2017) and Contrastive Language-Image Pretraining (CLIP) (Radford et al., 2021) to the problem of sentiment analysis in flooding situations, in order to create a more generalisable approach — both in terms of analysing new events, and in terms of handling new situations such as different places or languages.

3. Data

To collect data for analysis, we identified major flood and extreme weather events in the last ten years and collected the historical tweets that were related to them based on hashtags and/or keyword search using Twitter's Academic API. We used the following search terms to source tweets via the Twitter API: #StormDennis, #StormChristoph, #StormJorge, #HurricaneEta OR #HuricaneEta, #HurricaneBeta OR #HuricaneBeta, #HurricaneIota OR #HuricaneIota, #NSWFloods, #qldfloods, #ColoradoFloods OR #ColoradoFlooding, #MexicoFloods, flood #snaith, #floods OR #flashfloods, #StormFranklin, finsbury park flood, himachalpradesh flash floods, texas floods, #SydneyFloods. The tweets we collected span from 2007-12-06 to 2020-02-25.

Table 1 details the floods from which data was downloaded. The start and end date columns refer to the time frame of tweets appearing rather than to the events themselves. To preprocess the data, we excluded retweets and deduplicated tweets by their IDs (though some duplication is possible if multiple users copy the same text and post it independently). Replies to tweets matching the search criteria were included.

For all tweets collected, we kept information on: the search term itself, the tweet ID, user name and description, user location, the number of followers and tweets by a user, the number of retweets and likes of a tweet, any media (e.g. images), and the geolocation of the tweet, if provided. While we downloaded 13,851 tweets from the hashtag #StormFranklin (including replies), we excluded these from the training and validation datasets for later evaluation. This gave us a dataset of 795,065 tweets in total, including the StormFranklin tweets.

Fig. 1 shows some examples of tweets with images. Tweets 1(a) and 1(b) could be classed as positive ("be safe", "easy way to do it without getting wet!"), with nobody in immediate distress. Meanwhile, tweet 1(c) could be classed as negative ("frustrated and upset"), potentially highlighting an issue that requires human attention. Below some examples of tweets without images are shown:

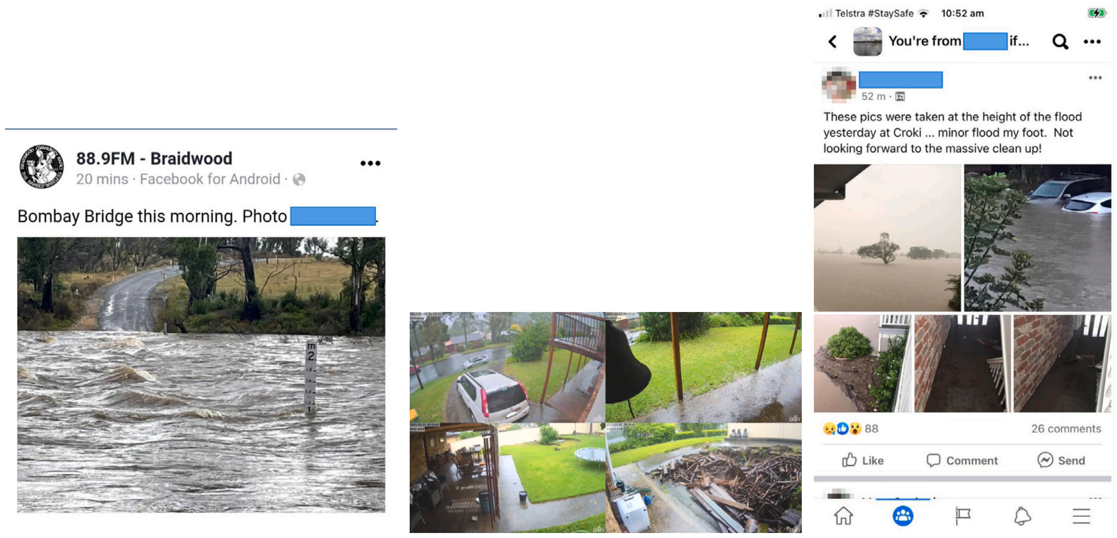
> "*@sZL7YcOsTnZhhzsf8xFXcA @JF1MQxDGoRT7NsObnSxQpA And it's still bucketing down in Coffs, landslips at the Big Banana and Thora, Waterfall Way. These hills are supposed to keep us high and dry but nooooo*" –19th March 2021

> "*@fXOQqWAYZqJlpv_r-qIy2 A @sZL7YcOsTnZhhzsf8xFXcA All good here Cows and calves on the hill paddock All other animals + humans safe*" –19th March 2021

> "*@jd2a0Qryu0aGHcz9bj4B2 A Extraordinary that Warragamba is full. It's a massive dam. Hope it goes ok for those downstream*" –20th March 2021

Tweet 2 here could be classed as positive, whereas tweet 1 would be negative as it indicates someone's house has been flooded. Tweet 3 could be classed as negative, as it contains information that could have a significant negative effect on those downstream.

Tweets were anonymised using the SHAKE128 hash function (Dworkin, 2015) with a salt. We hashed all tweet ids, usernames, and conversation ids. We kept geotags and place names extracted by twitter, along with direct links to associated media.


88.9FM - Braidwood

20 mins · Facebook for Android ·

Bombay Bridge this morning. Photo [redacted].

Telstra #StaySafe 10:52 am

You're from [redacted] if...

[redacted]

52 m ·

These pics were taken at the height of the flood yesterday at Croki ... minor flood my foot. Not looking forward to the massive clean up!

(a) Well and truly! Be safe everyone, don't drive thru floodwaters. #Braidwood #NSWFloods https://t.co/IZvGEvPQ7A https://t.co/2Frl4LdWlp

(b) Using my @q_UuWRrU0H0gdatqcI8Lqw #Protect cameras to keep an eye on house during the #NSWFloods. Easy way to do it without getting wet! https://t.co/DeSH6kDl8h

(c) And understandably people are frustrated and upset because they are trapped and weather & emergency descriptions are not always accurate as Mary from #croki near #Taree points out. #nswfloods #MidNorthCoast #flooddisaster https://t.co/yO1kwRxVTU

Fig. 1. Example tweets with images.

Positive: 😂👍🤣❤️👏😊😉😁😘😍🤗💕😀💙💪✅🙌💚👌🙂😎😆😅☺️😃😻💖💋💜
😹😜♥️😄💛😽✔️👋💗✨🌹🎉👊😋😏🌞😇🎶⭐💞😺😸🖤🌸💐🍀🌼🌟🤝🌷🐱🤓
😌😛😙

Negative: 🥶⚠️💔😰🙄😱😭😳😢😬🤦😡😩🙈😔☹😮❌😣😕😲😥😞😫😟😴🙀😒🙁
😠😪😯😨👎🤢💀😤😐😖😝😈😑😓😿😵😧😶😦🤥🤐🤧

Fig. 2. The emojis and the categories they were manually assigned to.

3.1. Data labelling

The learning models we developed and trained require labels for the training process. To this end we were inspired by Felbo et al. (2017) and extracted a list of all unique emojis from the dataset, which we then manually labelled to be either positive or negative — see Fig. 2. By positive and negative, we define positive to be tweets of no concern, and negative to be tweets potentially requiring attention (i.e. where someone may require assistance).

Then, we extracted all tweets from the data that had at least one (positive or negative) emoji, and automatically labelled them as either positive or negative based on our categories. Whenever more than one emoji was present, the majority category was used. If an equal number of positive and negative emojis were present, the 'positive' category took precedence. Finally, we split the data into two parts, with 80% for training and 20% for validation during the learning process.

To evaluate the accuracy of our sentiment labels against a human gold standard, we used Amazon Mechanical Turk (AMT) to collect human sentiment labels on a representative data sample of 1938 tweets randomly selected from the #NSWFloods hashtag from 40 different raters. #NSWFloods was chosen as it is time-limited and has a significant sample size (~15K). Turkers were presented with the tweet text (excluding images and emojis) and asked to assign a categorical rating from a 1–5 Likert scale, where 1 = negative, 2 = slightly negative, 3 = neutral, 4 = slightly positive and 5 = positive. We collected fine-grained ratings to allow a better comparison with models such as Valence Aware Dictionary and sEntiment Reasoner (VADER) (Hutto and Gilbert, 2014) or RoBERTa who predict such distinctions. Accuracy was low for finer sentiment distinctions, so 1–2 were collapsed as "negative" and 3–5 were considered "positive". These tweets form the basis of our experiments. Table 4 provides a comparison of our learning models against the human gold standard annotations.

4. Approach

This section will introduce the data representation and learning models for our experiments.

4.1. Representation of inputs

In our experiments, we compared the effectiveness of different AI model architectures in predicting the sentiment of social media posts from Twitter. Specifically, we compare the following architectures:

- **Pretrained baselines:** VADER, RoBERTa.
- **Models we trained:** Transformer, LSTM, CLIP, ResNet50 (He et al., 2016).

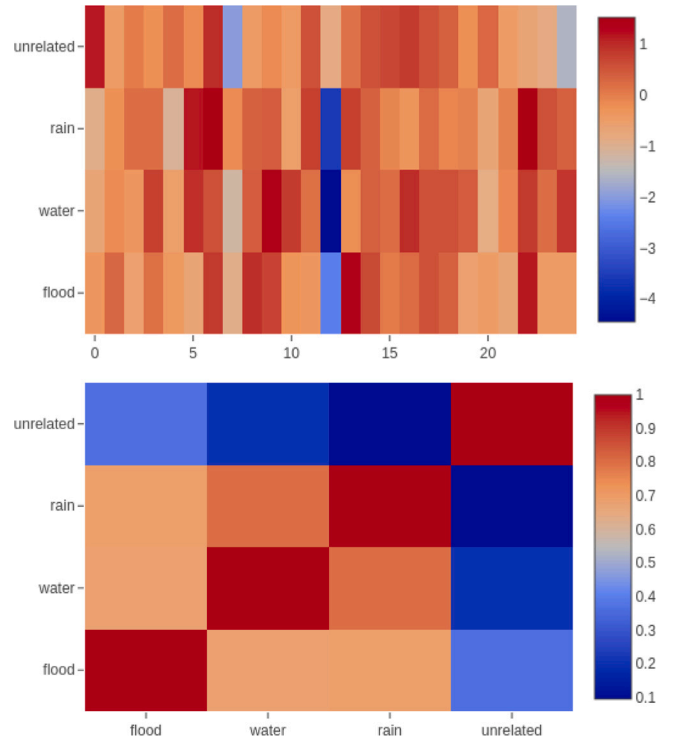


Fig. 3. Top: GloVe (trained on 2 billion tweets with vocabulary of 1.2 million words) word embedding vectors with a size (dimension) of 25 visualised as an array of numbers in a heatmap for a selection of words. Bottom: Using cosine difference, two vectors can be compared. `flood` and `rain` have a high similarity, whereas `flood` and `unrelated` have a low similarity. Such semantic comparisons are not possible e.g. using a one-hot encoding model — this enables the AI model that follows the embedding layer to focus on the domain-specific task, rather than having to learn not only the domain-specific task, but also relationships between words (Pennington et al., 2014).

When training models on natural language data, a problem is the size of the input data. e.g. if a dataset contains 15K unique words, using one-hot encoding a vector in the form $[word_0, word_1, word_2, \dots, word_{15000}]$ is required to represent each word, where $word_n \in 0, 1$. This has significant implications on both memory usage and generalisability – as one-hot encoding does not capture any lexical, semantic, syntactical meaning, or relationships – so alternate strategies are needed, such as Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014).

An embedding layer is often the first layer in an AI model, and it encodes the input data – which can in this case be defined as a string of text tokens from a tweet split up into its constituent words – into an array of numerical vectors of a fixed size. Such an embedding layer can be defined by a dictionary in the form $f : word^{1..n}_{string} \rightarrow vector^{dim}_{float32}$, which is then executed for every element of the input array of strings, where dim is the size of the resulting vector. Hence, an input to an embedding layer can be defined as $input^{words}_{string[]}$ (where $words$ is the number of words in the array of strings), and the output defined as $vector^{words,dim}_{float32}$.

Therefore, the memory required to encode the words from the original input is significantly reduced, while also ensuring that semantic meaning is retained by encoding words with a similar semantic meaning to similar numerical values (Koehrsen, 2018) (Fig. 3). These embeddings can be trained from the domain corpus, e.g. our tweets, but are typically trained separately from large general-purpose domains. They can be seen as a generalised representation of the English language, for example, comprised of different types of text, domains, and genres, e.g. GloVe (Pennington et al., 2014).

4.2. Learning models

The body of a neural network model follows the embedding layer in the form of a set of hidden layers.

Consider an input vector in of inputs that we want to map to a sentiment value out. To do this, we learn a hidden representation $h(in)$ using a function $f(h, in)$, minimising the loss (error) between a given label out^{truth} and predicted value $out^{predict}$ — e.g. using cross-entropy loss. We end up with a function that predicts a sentiment value as $out^{predict} = h(in)$

The first model we will train is the transformer network. A full transformer can be applied to e.g. machine translation tasks by translating one sequence of vectorised inputs into another. Transformers are made up of two parts: an encoder, which encodes features deemed important by the model into a sequence of vectors with a lower dimensionality (i.e. a feature map), and a decoder, which converts the feature map into the desired output.

The models used in this paper can be classed more specifically as deep feed forward networks with back propagation (Goodfellow et al., 2016a) - the process by which these models are trained using pairs of samples and associated labels. At each step of the training process:

1. The input sample is put through the model forwards through the directed graph of layers to make a prediction (feed-forward).
2. The prediction is compared to the ground truth label, and an error value is calculated using a loss function, for example mean squared error (i.e. $loss = (actual - predicted)^2$) and cross-entropy loss (e.g. $loss = -(actual \times log(predicted) + (1 - actual) \times log(1 - predicted))$) (Kaller, 2019; Brownlee, 2019).
3. Finally, the loss is propagated backwards through the model using an algorithm like gradient descent (Goodfellow et al., 2016b).

We use just the encoder part of a transformer to encode the vectorised input social media post into a feature map which can be interpreted by later layers of the model. Transformers handle sequences in parallel - as in $out_i = f(in_i)$. This is achieved by adding a positional embedding signal (explained below) and then dropping them through layer normalisations and dense layers, and a self-attention layer, which enables the model to identify which parts of the input are important for making the output prediction.

Alternatively, LSTMs (Hochreiter and Schmidhuber, 1997) or GRUs (Cho et al., 2014) can handle sequenced data. These are called recurrent models as they have a hidden state, and a system of gates are used to update the values therein for each element in the sequence, requiring that each element in a sequence is processed serially — e.g. $out_i = f(out_{i-1}, in_i)$.

This is why the transformer architecture enables greater computational parallelisation when training using a GPU (Vaswani et al., 2017), since without a recurrent element each sequence element can be processed independently, as in $out_i = f(in_i)$.

Transformers instead rely on dense (fully connected) layers. While this makes them more parallelisable and hence can train more quickly, this also means that they are unable to account for context and relative positioning in input sequences. To alleviate this weakness, a positional embedding code is added to the input sequence. If the input sequence (post-embedding layer) is $input^{seq,dim}$ (where seq is the sequence length and dim is the vector size for each element therein), then the positional embedding can be defined as (Vaswani et al., 2017):

$$PE_{(i_{seq}, 2i_{dim})} = sin(\frac{i_{seq}}{10000^{2i_{dim}/dim}})$$

$$PE_{(i_{seq}, 2i_{dim}+1)} = cos(\frac{i_{seq}}{10000^{2i_{dim}/dim}}),$$

...where i_{seq} is the position in the sequence dimension, i_{dim} is the position in the embedding dimension, and PE is the positional embedding for a single value in the $input^{seq,dim}$. The embedding dimension alternates between sin and cos for each successive element, as Fig. 4 shows.

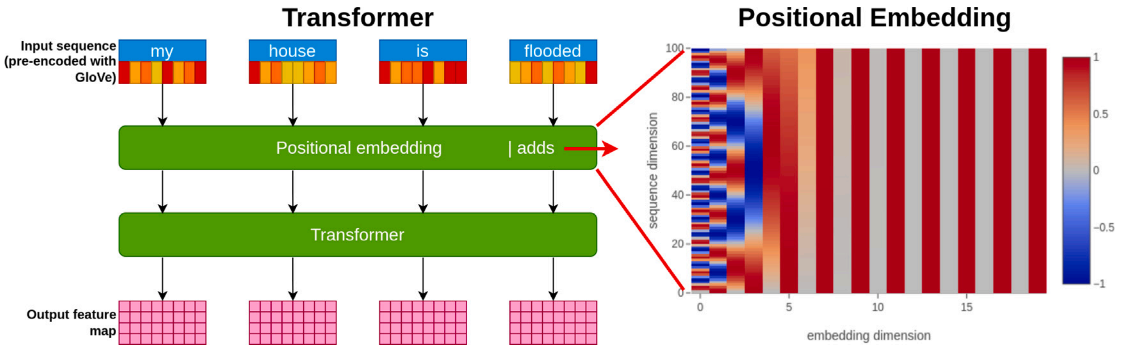


Fig. 4. The architecture of a transformer. The input sequence (that has already been encoded with GloVe as described in Section 4.1) gets a positional embedding formula added to it (visualised on the right; words in the sequence are along the vertical axis, and the embedding of those words are along the horizontal axis), which gives the information about the ordering of items in the sequence. Finally, the transformer itself processes it in parallel.

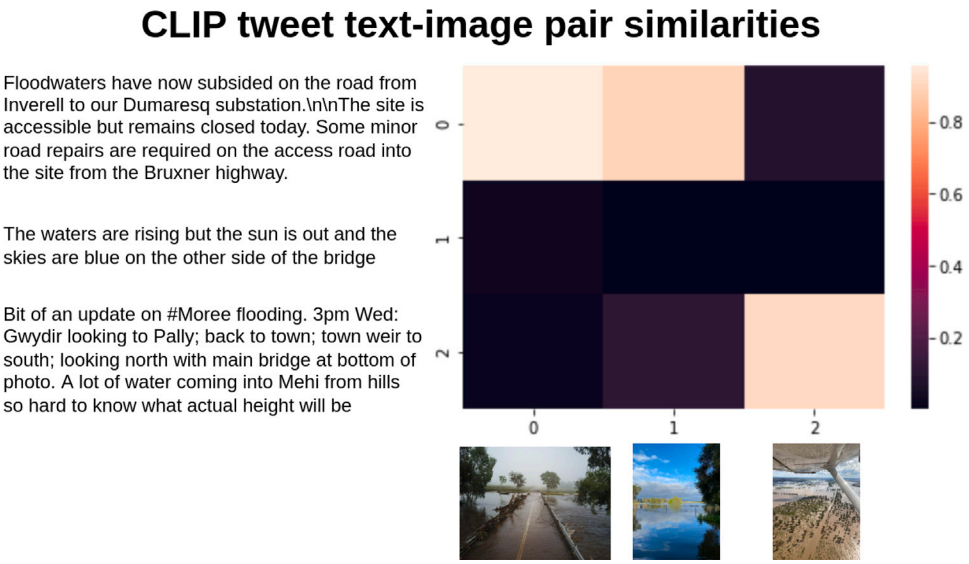


Fig. 5. Tweet text–image pairs and their similarities, as calculated by CLIP (higher values mean more similar). Although it is correct most of the time, as shown here sometimes the wording of the tweet confuses CLIP. Where multiple images associated with a tweet, only the first one was chosen.

4.3. Representing text and images jointly

With many users posting many tweets independently, many different topics are discussed in an unstructured manner. In some cases, images associated with tweets contain additional information as illustrated in Fig. 1. To explore the effect of images associated with tweets on sentiment analysis tasks, another model architecture we used in our comparison is the Contrastive Language-Image Pretraining (CLIP) architecture.

Similarly to the transformer already discussed, CLIP uses an embedding layer as its first layer. However unlike the plain transformer model, CLIP handles not only textual input but images as well. CLIP first has separate encoding layers for textual and image inputs, before later combining them together and training the two encoders to predict which textual and which image inputs were paired with each other. In doing so, CLIP trains to predict how well a textual string and an associated image pair together (Radford et al., 2021) (Fig. 5).

Fig. 6 outlines how CLIP trains and makes predictions by taking a contrastive learning approach. Batches of text–image pairs are compared using cosine similarity, which is used to calculate cross-entropy loss (see Fig. 7).

5. Experiments

This section presents the experimental setup we adopted for our experiments, presents and discusses results as well as some sample predictions.

5.1. Experimental setup

We compare a set of different methods for predicting sentiments from tweets given our emoji-based labels from Section 3. Specifically we compare:

1. **LSTM:** 2 bidirectional layers, 128 units each, batch normalisation.
2. **Transformer:** 1 transformer encoder, 16 attention heads, 32 units, dropout 0.1, gelu.
3. **CLIP:** Pretrained, ViT-B/32, followed by 2 × 512 unit dense layers, dropout 0.1.
4. **ResNet50:** ResNet50 architecture, followed by a softmax dense layer.

The hyperparameters of these models were chosen after experimentation with different combinations. In all cases where we trained a model, we used the Adam optimiser.

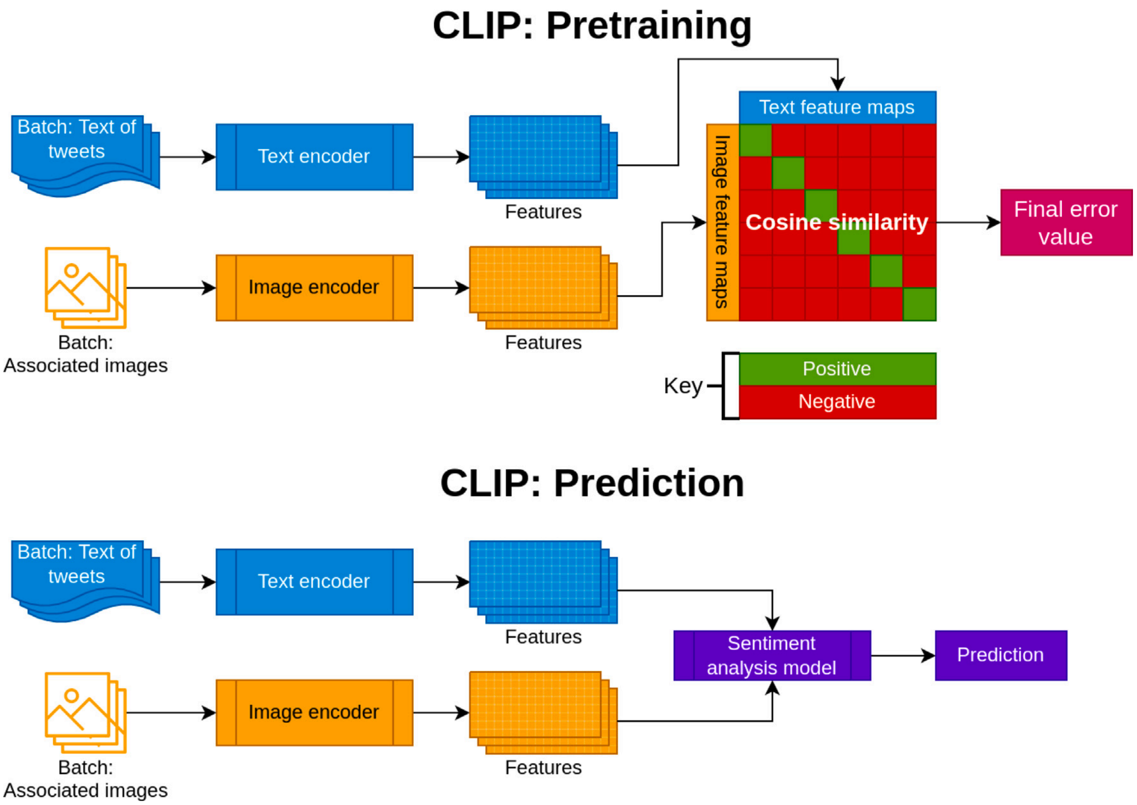


Fig. 6. A summary of how the CLIP model operates. A batch of text and image pairs are run through different encoders separately to produce a pair of feature maps. Then, the resulting feature maps are compared using cosine similarity and fed into the loss function which trains the model to learn pairs to be similar to one another (Radford et al., 2021). When making a prediction, the trained encoders can be used to encode new text–image pairs that are fed into a domain-specific model.

```
Inputs:
$image\_embedded^{batch\_size,dim\_embed}$
$text\_embedded^{batch\_size,dim\_embed}$
$batch\_size$
$t$: Learned temperature parameter
Algorithm:
$logits \leftarrow np.dot(image\_embedded, transpose(text\_embedded)) \times np.exp(t)$
$labels \leftarrow np.arange(batch\_size)$
$loss\_image \leftarrow cross\_entropy\_loss(logits, labels, axis = 0)$
$loss\_text \leftarrow cross\_entropy\_loss(logits, labels, axis = 1)$
$loss = \frac{(loss\_image+loss\_text)}{2}$
```

Fig. 7. An outline of CLIP's contrastive learning loss algorithm.
Source: Adapted from Radford et al. (2021).

We also chose two baseline models for comparison that were shown to perform well on the task of sentiment analysis in previous work: (1) VADER (Hutto and Gilbert, 2014), a rule-based model developed to predict the sentiment of social media posts, and (2) RoBERTa (Liu et al., 2019), a generic pre-trained transformer based model.

We trained our models on various Nvidia GPUs: GeForce 3060, Tesla K40m, Tesla P100, and Nvidia A40, depending on availability and machine learning library requirements (see Table 2).

Accuracies are reported from models with the same architecture. All models were trained for a total of 50 epochs, and then the checkpoint from the epoch with the highest validation accuracy was chosen. All models (except CLIP, which has its own inbuilt word embeddings) also used GloVe pretrained on Twitter data with a dimension of 200 for word embeddings as it is more computationally efficient, although other word embeddings do exist (Liu et al., 2019; Lewis et al., 2020; Devlin et al., 2019). We test the potential of this technique against our RoBERTa and VADER baselines.

Table 2
An overview of the models used.

Model	Parameters	Validation accuracy
Pretrained RoBERTa	345M	n/a
LSTM	731K	81.81%
Transformer	2.6M	80.42%
CLIP (not augmented)	152.1M	89.41%
CLIP (augmented)	152.1M	86.37%
ResNet50	23.6M	73.79%

The transformer encoder (Vaswani et al., 2017) model we trained predicts the positive/negative sentiment of the tweet text, using the emojis as labels as described in Section 3. Although emoji-based labels (see Section 3.1) are required during the training process, emojis are not required during inference. Fig. 8 shows the architecture of the transformer encoder model we trained, as well as that of the CLIP-based model.

Table 3
Comparison of the sentiment analysis performance against emojis as a ground-truth label.

Model	F1	Recall	Precision	Accuracy	Samples	Truth Pos	Predict Pos	Truth Neg	Predict Neg
VADER	0.339	0.596	0.506	0.471	15,485	7183	15,224	8302	261
RoBERTa	0.651	0.658	0.651	0.658	15,486	7183	6122	8303	9364
Transformer	0.687	0.694	0.693	0.687	15,486	7183	8674	8303	6812
LSTM	0.676	0.72	0.695	0.682	15,486	7183	10,381	8303	5105
CLIP (augmented)	**0.802**	**0.791**	**0.817**	**0.841**	2491	641	738	1850	1753
CLIP (not augmented)	0.763	0.752	0.814	0.792	2491	641	978	1850	1513
ResNet	0.697	0.691	0.742	0.734	2315	585	918	1730	1397

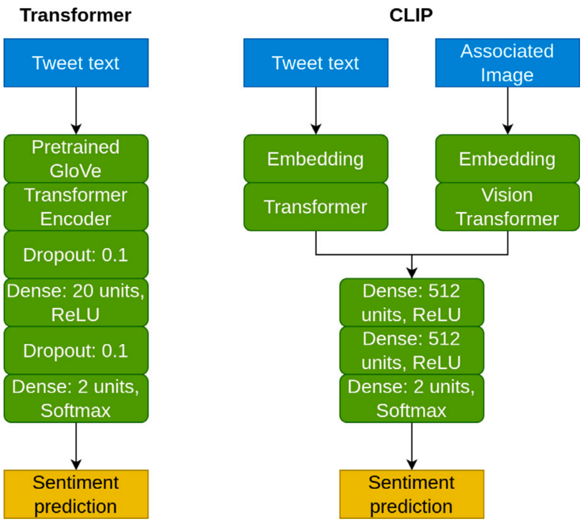


Fig. 8. The architectures of our transformer encoder (left) and CLIP (right) models that we trained. The CLIP model concatenates the feature maps from both the image and the text encoders.

Like the transformer, labels for training the CLIP model came from emojis split into positive/negative categories. The CLIP model takes both text and images as an input at the same time before then producing a prediction based on both inputs. Fig. 8 shows the architecture of the CLIP model trained. We discovered that out of the 180K tweets that had an associated image, only 14K tweets also contained an emoji (i.e. an output label according to our setup).

To augment the dataset, we used the CLIP model trained on 14K image–text pairs to annotated each tweet that had an emoji but no image with a newly associated image that fits the text. Fig. 9 shows the algorithm that we used to augment the data. This augmentation process raised the size of the training dataset to 55K text–image pairs, and improved the F1 score of the model from **76.3%** to **80.2%** (see Table 3). Tweets without associated images achieved 0.734 F1 (CLIP-augmented)/0.766 (CLIP-not augmented), and tweets with images 0.8 F1 (CLIP-augmented)/0.767 (CLIP-not augmented).

5.2. Results

Comparison against emoji labels. To compare the performance of the models, we used the emoji labels, as we used previously to train the CLIP and transformer models, as a ground truth. Table 3 shows the results of this experiment. We used the 15K tweets from the `#NSWFloods` hashtag for this emoji-based comparison and any predictions of `neutral` were considered `positive` predictions instead, as detailed in Section 3.

Comparison against human ratings. To further explore the comparative performances of these models, we also used our random human-labelled subset of 1938 tweets from the `#NSWFloods` hashtag and analyse the performance of our models against them — Table 4 shows the results of this experiment. As in Table 3, predictions with a class of `neutral` were considered `positive`. These results show that – unlike with the emoji labels in Table 3 – our transformer model is the best performing model, with the CLIP and LSTM models coming in second place. We speculate this could be because emojis and images were not used in the human labelling process or due to the small sample size.

Images-only results. Our ResNet50 model took images associated with the tweets as an input and classified them as positive or negative, using labels predicted by the LSTM model from the associated text as a ground-truth. As this model analyses images rather than text, only a subset of the tweets in Tables 3 and 4 could be used for the ResNet50 model.

5.3. Discussion

Of all the models we tried, under F1 score our CLIP model appears to perform best, followed by our ResNet model when compared against emojis as a ground truth in Table 3, while our transformer model comes in a close third. Linguistic clues can sometimes be quite subtle (e.g. sarcasm), often requiring the reader to infer the correct meaning. Deep learning models find this challenging, so images can help fill this gap. Tweets T1 and T2 in Table 5 are examples of this, where the image provides critical context that is otherwise misinterpreted by other models.

Our baseline model VADER appears to perform worst. We suggest that this may be because emojis play a key role in identifying the sentiment of tweets and VADER does not adequately take the context in which an emoji is used into account, and its rule based approach is inflexible when compared to models that are trained in a supervised or semi-supervised manner. This is illustrated by tweets T3 and T4 in Table 5 — which although it is a negative tweet, it is still considered neutral by VADER. This is also backed up by the human-labelled tweets, and an accuracy of 42% (human-labelled)/33.9% (emoji-labelled).

When compared using human-labelled tweets as a ground truth instead, the story is very different. All scores are generally lower, indicating that the smaller size of the human-labelled dataset may not be completely representative of the entire dataset. Despite this, our Transformer performs best and RoBERTa performs worst, suggesting that the MultiNLI dataset that the RoBERTa model was trained on (Lewis et al., 2020) is not representative of the target domain of social media here.

Humans are better at inferring meaning in language than AI, but with visual information not being present for our human raters this makes the task more challenging. Given the human-labelled dataset was small (~2K tweets) and human-labelling tweets is both time consuming and expensive, labelling tweets automatically via emojis is significantly more practical.

5.3.1. Sentiment by image

Since images associated with tweets can contain some useful information (Said et al., 2020), we used CLIP to explore utilising both text and images to predict sentiment in order to understand how images relate to the overall sentiment of a tweet.

With 149.5M more parameters than our Transformer model (approx. 788K of which are in dense layers we trained after the pretrained

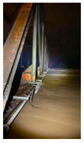
```
for(tweet of dataset) {
      rankings = [];
      for(image of images) {
            rankings.push(clip.rank_image(image));
      }

      high_rankings = rankings.filter(ranking => ranking > 0.75)
      if(high_rankings.length > 0)
            return random_item(high_rankings);
      else
            return highest_ranking(rankings);
}
```

Fig. 9. The algorithm by which we ranked image associations with tweet texts when augmenting the tweets with CLIP.

Table 4
Comparison of the sentiment analysis models we tried against human-labelled tweets.

Model	F1	Recall	Precision	Accuracy	Samples	Truth Pos	Predict Pos	Truth Neg	Predict Neg
VADER	0.499	0.501	0.501	0.566	1914	1358	1258	556	656
RoBERTa	0.42	0.521	0.521	0.421	1914	1358	517	556	1397
Transformer	**0.591**	0.595	0.613	0.623	1914	1358	1095	556	819
LSTM	0.589	0.587	0.595	**0.645**	1914	1358	1263	556	651
CLIP (augmented)	0.512	0.604	0.602	0.512	320	220	96	100	224
CLIP (not augmented)	0.58	**0.607**	**0.624**	0.588	320	220	144	100	176
ResNet	0.54	0.559	0.576	0.577	156	116	84	40	72

Table 5
Some sample tweets from the #NSWFloods dataset labelled by the various models we tested. The Column CLIP refers to the augmented model.

	Text	Emoji	Human	Transformer	LSTM	VADER	RoBERTa	ResNet	CLIP
T1	Hastings river port Macquarie #NSWFloods	n/a	neutral	positive	negative	neutral	negative	negative	negative
T2	😂 The local Facebook page is "delivering" today	positive	n/a	negative	negative	neutral	positive	positive	positive
T3	Droughts.. Fires.. Floods.. #Australia #NSWFloods #SydneyFloods Oh and a bit of #COVID19 Wasn't 2021 meant to be a better year?	n/a	negative	negative	positive	neutral	positive	n/a	negative
T4	Before and after pics of Wauchope railway bridge 😳 #NSWFloods credits to #<name redacted> 	negative	negative	negative	negative	neutral	negative	negative	negative

CLIP model), it is also significantly more computationally expensive and may have overfit. The tweet augmentation process is especially computationally expensive, requiring each tweet to be ranked against every image in the dataset.

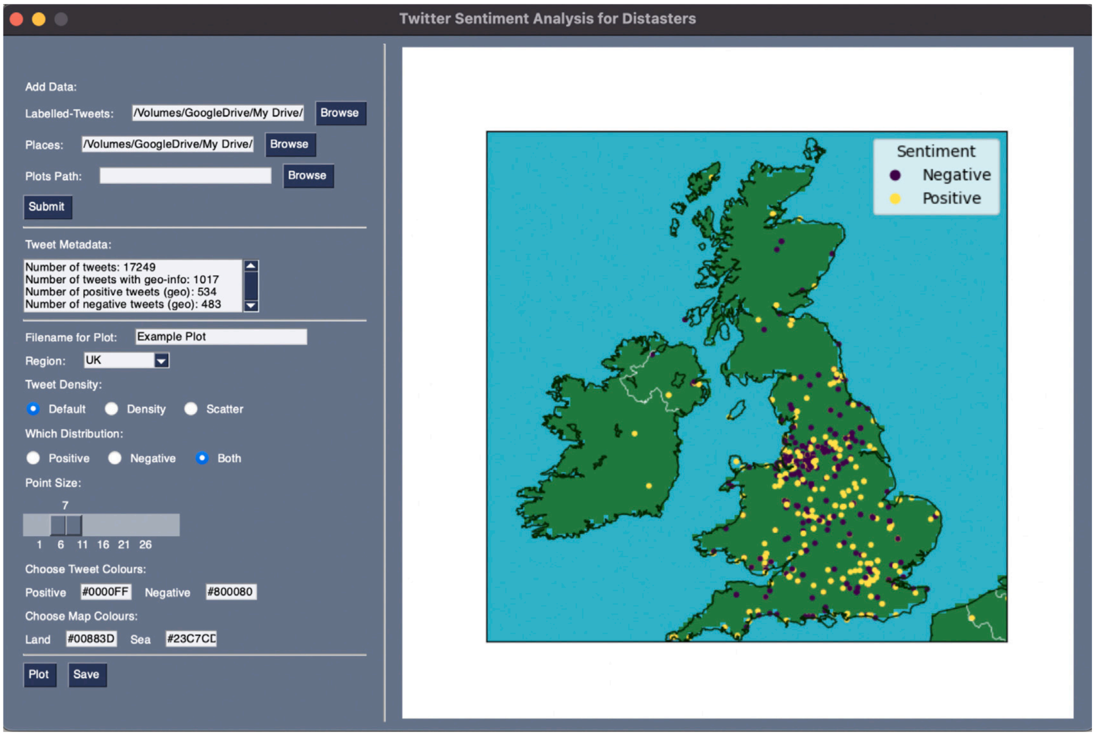
When compared to emojis as a ground-truth label, our CLIP model easily beats the all the other models that consume only textual data by a significant margin of at least 11%. However, when we compare it to human-labelled tweets, it does not outperform the text-only transformer even though CLIP also had associated images as an input.

This illustrates that images associated with tweets contain contextual information that was lost to human raters. Given the small sample size mentioned earlier, this further shows that it is more practical to use emojis as labels, and to include images for additional visual context.

To further explore the relationship between images and sentiment, we can look to our image-only ResNet model, which appears to be the highest performing model in Table 3 with respect to both F1 score and accuracy. This may be due to a small sample size used in the comparison as the ResNet only analyses tweets containing at least one image (2315 samples vs 15,486 samples for the transformer), and the relatively unbalanced dataset as compared to the text-only models — suggesting that when people tweet image(s), these images are more likely to be considered to have a negative sentiment.

5.3.2. *Geospatial analysis of tweets*

To further improve upon the explainability of the flood sentiment analysis, geo-spatial analysis was performed using tweets sentiment-analysed by our Transformer model. Social media systems track the locations of users, thereby making it possible to determine where a person was when they posted on the site or application.



Add Data:

Labelled-Tweets: /Volumes/GoogleDrive/My Drive/

Places: /Volumes/GoogleDrive/My Drive/

Plots Path:

Tweet Metadata:

```
Number of tweets: 17249
Number of tweets with geo-info: 1017
Number of positive tweets (geo): 534
Number of negative tweets (geo): 483
```

Filename for Plot: Example Plot

Region: UK

Tweet Density: ● Default ○ Density ○ Scatter

Which Distribution: ○ Positive ○ Negative ● Both

Point Size: 7 (1 6 11 16 21 26)

Choose Tweet Colours: Positive #0000FF Negative #800080

Choose Map Colours: Land #00883D Sea #23C7CD

Fig. 10. Graphical user interface for the geo-spatial plotting system. Tweets can be optionally grouped by date, allowing for temporal analysis of geospatial trends.

Table 6
A sample of tweets from 2022-02-18 for the hashtag #StormFranklin, from before the storm actually hit. In total, there are 11 positive and 24 negative tweets. Tweets classed as positive generally make light of the storm (P1, P2, P3, P4), while negative tweets are either sarcastic (N1) or worrying about causes and effects (N2, N3). Additionally, 5 of the positive tweets are written in the Dutch language (Google Translate used for identification), but were still classified correctly (P4).

	Text	Label
P1	Before retiring to bed this evening, folks, don't forget to make sure your beer's properly tied down #StormEunice #StormFranklin #Beer #BeerBods	positive
P2	#StormFranklin on Sunday so don't give the neighbours back fences.wheelie bins and trampolines from #stormeunice just yet"	positive
P3	@cIAD_xsnxg-KiLUkM8tK6 g @oB2hntIJShfk6tmiZZ7z8Q The next storm will begin with a F. (or is that an F??) They're calling it #StormFranklin I'm looking forward to #StormInATeacup 🤣 🤣 🤣	positive
P4	Ik ben benieuwd wat Eunice ons gaat brengen! Stay strong! #eunice #storm #StormEunice #StormFranklin	positive
N1	Woohoo! #StormFranklin is due after #StormEunice takes her hook 🙈	negative
N2	How much do these extreme storms make you worry about climate change?#StormEunice #StormDudley #StormFranklin	negative
N3	The storm coming in Sun/Mon #StormFranklin could even more concerning than Eunice. Keep your eyes peeled. Concern is it will arrive only a day or so after Eunice. #windy #storm #StormEunice	negative

We developed a program to create geo-spatial visualisations from geotagged tweets to explore large datasets (shown in Fig. 10).

There are two types of geographical metadata available for tweets from the Twitter API: the exact location of the origin of the tweet (from GPS-enabled or GeoIP-enabled devices), and the Place object type extracted from the tweet text content using named-entity-recognition which provides a polygon which bounds the area from which the user posted the tweet.

We used the exact coordinates where possible, and took the central point of the polygon bounding-box for tweets without precise location data. To account for the overlap of points representing tweets from the same Place region, two distinct methods were employed: increasing the size of the points as a function of the number of tweets contained within, and adding Gaussian noise to points superimposed on a single coordinate.

We used tweets from the #StormFranklin hashtag (see also Table 1) for our geospatial analysis. This extreme weather event was chosen as a large number of tweets were available and it was not included in any data downloaded previously (Hurricane Iota was downloaded separately). Our Storm Franklin dataset has 921 geotagged tweets (7%), and only 28 include precise location information. Sloan and Morgan (2015) showed that approximately 0.85% of tweets are geotagged, but Twitter made changes in 2019 to reduce the ways by which precise location geotagging can be achieved, explaining the proportions found (Hu and Wang, 2020). 29% of the tweets were classified as having positive sentiment.

The static plot shows that the vast majority of negative tweets are found in high-density clusters, significantly more so than the positively labelled tweets. This is most obvious for tweets originating in Ireland.

Fig. 11 shows the temporal-level distribution of tweets for Storm Franklin. Tweet sentiments are visualised from 2022-02-18 to 2022-02-25, and a time-series of tweet frequency - the majority of tweets were posted over a two day period: 2022-02-19 to 2022-02-20.

Days with most activity correspond with days with the highest number of tweets posted, and is also slightly before the actual event, which took place on 2022-02-20 to 2022-02-21 (Deltares, 2022). This could be because the Met Office's early warning prompted conversation on Twitter or because of lingering effects from Storm Eunice, which took place a few days prior - see Table 6 (Deltares, 2022).

By analysing sentiment-analysed tweets geospatially, real-time information on the status of flooding events can be obtained. By analysing the effectiveness several sentiment analysis models in Section 5, we improve our geospatial analysis of sentiment-analysed tweets. Similarly, by mapping tweets we gain insights into the effectiveness of our sentiment analysis model.

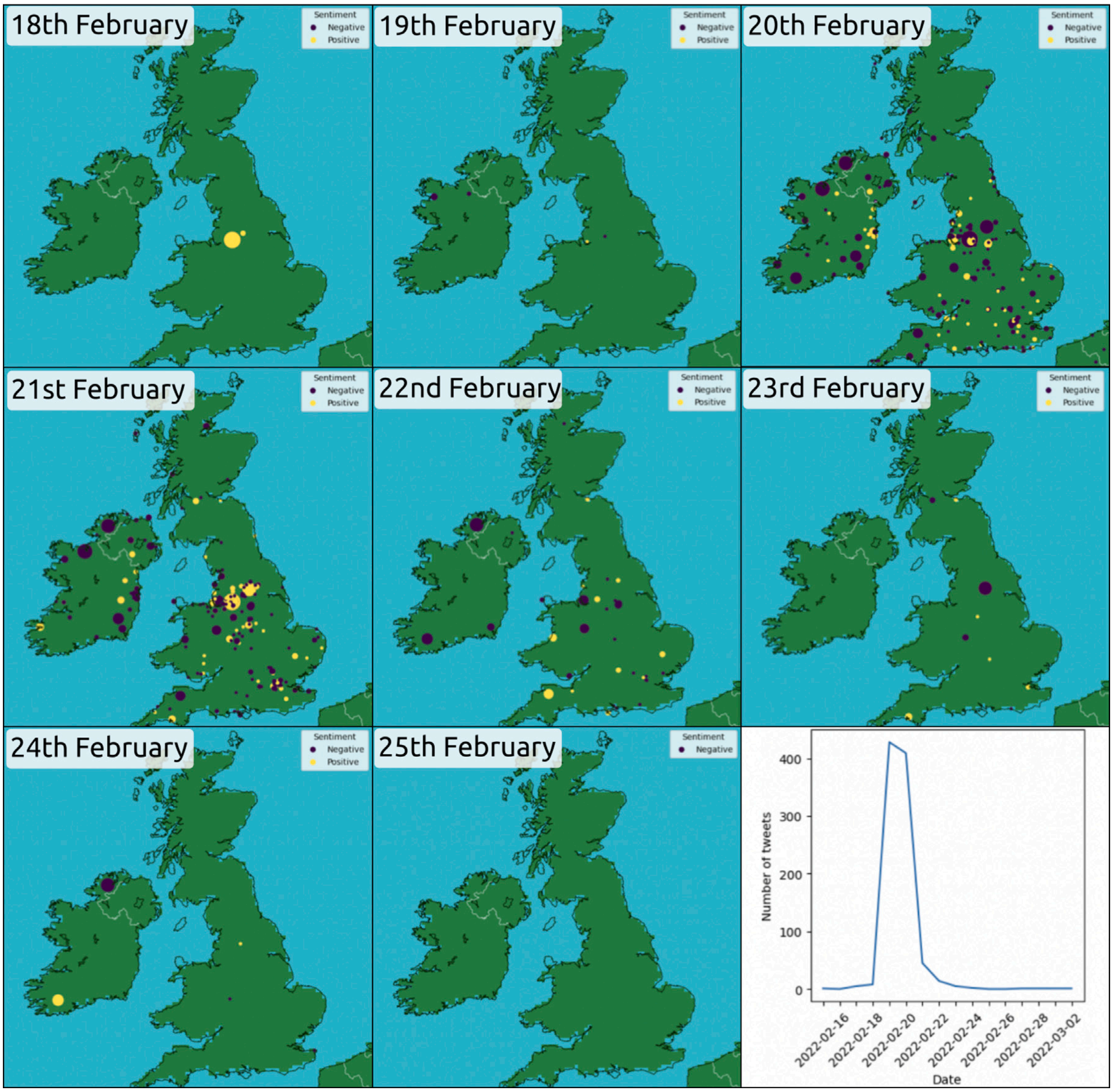


Fig. 11. Daily plots of geographically located sentiment-labelled tweets for the #StormFranklin dataset on 18th–25th February 2022.

6. Conclusion

We explored using text and images from Twitter for sentiment analysis, comparing VADER, RoBERTa, a Transformer encoder, and CLIP. Our CLIP-based model takes both text and images into account, which by taking visual information into account can help close the gap between AI and human raters. We highlighted the importance of emojis in understanding the sentiment of tweets: both our CLIP and transformer models outperformed RoBERTa and VADER, neither of which consider emojis.

Finally, we demonstrate the potential and feasibility of geospatial analysis of sentiment-analysed tweets in a flooding situation. The scale, frequency and low-latency allow for rapid analysis of disasters in real-time, enhancing situational awareness and allowing a human-lead approach to identification of affected areas in real time.

- An assessment of how our approach could be used to practically support disaster decision making in-situ is required.
- An alternative approach such as image segmentation (Pally and Samadi, 2022) or captioning is likely needed to make better use of images in the context of flooding events.
- Scope exists for future research to investigate models that consider emojis as well as regular text when predicting sentiment at inference.
- Correcting for population density, mass evacuations of people, and potential communication disruption are all challenges. Making use of other data modalities such as satellite data, mobile phone cell tower information, and traffic data may help here.
- Screening tweets from irrelevant and automated sources remains challenging.
- The relationship between social media response and flood severity is difficult to study given no consistent metric could be found that is not limited by country borders. For example, media reporting may have an effect on social media responses.

CRediT authorship contribution statement

Lydia Bryan-Smith: Conceptualization, Methodology, Data curation, Software, Investigation, Writing – original draft. **Jake Godsall:** Geospatial tweet analysis methodology, Analysis, Visualisation. **Franky George:** RoBERTa methodology, Software. **Kelly Egode:** VADER methodology, Software. **Nina Dethlefs:** Primary supervision, Writing – review & editing, Conceptualization. **Dan Parsons:** Conceptualization, Secondary supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Lydia Bryan-Smith reports financial support was provided by University of Hull. Jake Godsall, Franky George, and Kelly Egode reports financial support was provided by Natural Environment Research Council. Lydia Bryan-Smith reports equipment, drugs, or supplies and statistical analysis were provided by University of Hull Viper High Performance Computing facility. Lydia Bryan-Smith reports a relationship with University of Hull that includes: employment, funding grants, and non-financial support. Nina Dethlefs reports a relationship with University of Hull that includes: employment. Dan Parsons reports a relationship with University of Hull that includes: employment. Dan Parsons reports a relationship with Loughborough University that includes: employment. Lydia Bryan-Smith receives a PhD scholarship from University of Hull. Jake Godsall, Franky George, and Kelly Egode are students at the University of Hull. The NERC discipline-hopping grant declared was for a hackathon they attended. Dan Parsons was employed by the University of Hull, but has recently moved to Loughborough University.

Data availability

The authors do not have permission to share the original data as per the Twitter T&Cs. However, model checkpoints etc will be provided on request.

Acknowledgements

Lydia Bryan-Smith is funded by a PhD stipend from the University of Hull. We acknowledge the VIPER high-performance computing facility of the University of Hull and its support team. Some of the results presented in this article were developed during a Hackathon on Sustainable AI, hosted at the University of Hull, and funded by a NERC Discipline Hopping grant.

Appendix. Code availability

The code written in support of this paper has been published on GitHub. The following repositories contain the code in question:

- https://github.com/sbrl/twitter-academic-downloader (Mozilla Public Licence 2.0): The command line program written to download the tweets from Twitter, using Twitter's Academic API.
- https://github.com/sbrl/research-smflooding (GNU Public Licence 3.0): The code written to train and interact with the AI models tested in this paper.
- https://github.com/jakegodsall/twitter-floods (GNU Public Licence 3.0): The code written to geolocate and plot the sentiment of tweets on a map.
- https://huggingface.co/siebert/sentiment-roberta-large-english: The code used for sentiment analysis with RoBERTa.

Please note that all of these code repositories use other external open-source libraries to provide some functionality. For example, Tensorflow (TensorFlow Contributors, 2019) is used as an machine learning framework. All open source libraries used are open-source, defined in either `requirements.txt` (Python) or `package.json` (Node.js) freely downloadable from either PyPi (Python Software Foundation, 2022) (Python libraries) or npm (npm Inc, 2020) (Javascript).

Python was used as the main programming language. Javascript (Node.js OpenJS Foundation, 2020) was also used to initially download the tweets and to manipulate the data. Bash (shell scripting) was used in the analysis of the data.

References

Agüero-Torales, M.M., Salas, J.I.A., López-Herrera, A.G., 2021. Deep learning and multilingual sentiment analysis on social media data: An overview. Appl. Soft Comput. 107, 107373.

Arthur, R., Boulton, C.A., Shotton, H., Williams, H.T.P., 2018. Social sensing of floods in the UK. PLoS One 13.

Avvenuti, M., Cresci, S., Polla, M.N.L., Marchetti, A., Tesconi, M., 2014. Earthquake emergency management by social sensing. In: 2014 IEEE International Conference on Pervasive Computing and Communication Workshops. PERCOM WORKSHOPS, pp. 587–592.

Baccianella, S., Esuli, A., Sebastiani, F., 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: LREC.

Beigi, G., Hu, X., Maciejewski, R., Liu, H., 2016. An overview of sentiment analysis in social media and its applications in disaster relief. Sentiment Anal. Ontol. Eng. 313–340.

Brownlee, J., 2019. A gentle introduction to cross-entropy for machine learning. Available online: https://machinelearningmastery.com/cross-entropy-for-machine-learning/. (Accessed 12 October 2020).

Cho, K., van Merrienboer, B., Çaglar Gülçehre, Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: EMNLP.

Cortes, C., Vapnik, V.N., 1995. Support-vector networks. Mach. Learn. 20, 273–297.

Coulthard, T.J., Neal, J.C., Bates, P.D., Ramirez, J., de Almeida, G.A.M., Hancock, G.R., 2013. Integrating the LISFLOOD-FP 2D hydrodynamic model with the CAESAR model: implications for modelling landscape evolution. Earth Surf. Proces Landforms 38 (15), 1897–1906.

Davies, G., Roberts, S.G., 2015. Open source flood simulation with a 2D discontinuous-elevation hydrodynamic model.

Deltares, 2021. Home - DLEFT3D. Available online: https://oss.deltares.nl/web/delft3d. (Accessed 12 July 2022).

Deltares, 2022. Red weather warning issued for storm eunice - met office. Available online: https://www.metoffice.gov.uk/about-us/press-office/news/weather-and-climate/2022/red-weather-warning-issued-for-storm-eunice. (Accessed 18 July 2022).

Deltares, 2022. Storm franklin named - met ofice. Available online: https://www.metoffice.gov.uk/about-us/press-office/news/weather-and-climate/2022/storm-franklin-named. (Accessed 18 July 2022).

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

Dworkin, M., 2015. SHA-3 standard: Permutation-based hash and extendable-output functions.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., Lehmann, S., 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In: EMNLP.

FitzGerald, G., Toloo, G.S., Baniahmadi, S., Crompton, D., Tong, S., 2019. Long-term consequences of flooding: a case study of the 2011 queensland floods. Austr. J. Emerg. Manag. 34, 35–40.

Fu, X., Liu, W., Xu, Y., Yu, C., Wang, T., 2016. Long short-term memory network over rhetorical structure theory for sentence-level sentiment analysis. In: ACML.

Furquim, G., Filho, G.P.R., Jalali, R., Pessin, G., Pazzi, R.W., Ueyama, J., 2018. How to improve fault tolerance in disaster predictions: A case study about flash floods using IoT, ML and real data. Sensors 18 (3), 907.

Gao, H., Barbier, G., Goolsby, R., 2011. Harnessing the crowdsourcing power of social media for disaster relief. IEEE Intell. Syst. 26 (3), 10–14. http://dx.doi.org/10.1109/MIS.2011.52.

Goodfellow, I., Bengio, Y., Courville, A., 2016a. Deep Learning. MIT Press, pp. 164–223, Available online http://www.deeplearningbook.org. (Accessed 9 October 2020).

Goodfellow, I., Bengio, Y., Courville, A., 2016b. Deep Learning. MIT Press, pp. 271–325, Available online, http://www.deeplearningbook.org. (Accessed 9 October 2020).

Gould, I., Wright, I., Collison, M., Ruto, E., Bosworth, G., Pearson, S., 2020. The impact of coastal flooding on agriculture: A case-study of lincolnshire, United Kingdom. Land Degrad. Develop. 31, 1545–1559.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 770–778.

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

Hu, Y., Wang, R.-Q., 2020. Understanding the removal of precise geotagging in tweets. Nat. Hum. Behav. 4 (12), 1219–1221.

Hutto, C.J., Gilbert, E., 2014. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In: ICWSM.

Janowicz, K., Gao, S., McKenzie, G., Hu, Y., Bhaduri, B., 2020. GeoAI: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond. Int. J. Geogr. Inf. Sci. 34 (4), 625–636. http://dx.doi.org/10.1080/13658816.2019.1684500.

Kaller, J., 2019. Loss functions in machine learning for beginners |by John Kaller |AI^3 |theory, practice, business |medium. Available online: https://medium.com/ai3-theory-practice-business/loss-functions-in-machine-learning-for-beginners-fastai-lesson-9-homework-2-2121954f1f77. (Accessed 09 October 2020).

Kankanamge, N., Yigitcanlar, T., Goonetilleke, A., Kamruzzaman, M., 2020. Determining disaster severity through social media analysis: Testing the methodology with south east queensland flood tweets. Int. J. Disaster Risk Reduct. 42, 101360.

Keung, K.L., Lee, C.K.M., Ng, K.K.H., Yeung, C.K., 2018. Smart city application and analysis: Real-time urban drainage monitoring by IoT sensors: A case study of Hong Kong. In: 2018 IEEE International Conference on Industrial Engineering and Engineering Management. IEEM, pp. 521–525.

Khayyam, U., Noureen, S., 2020. Assessing the adverse effects of flooding for the livelihood of the poor and the level of external response: a case study of hazara division, Pakistan. Environ. Sci. Pollut. Res. 27, 19638–19649.

Koehrsen, W., 2018. Neural network embeddings explained |by will koehrsen |towards data science. Available online: https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526. (Accessed 05 May 2022).

Kokab, S.T., Asghar, S., Naz, S., 2022. Transformer-based deep learning models for the sentiment analysis of social media data. Array.

Kongthon, A., Haruechaiyasak, C., Pailai, J., Kongyoung, S., 2012. The role of Twitter during a natural disaster: Case study of 2011 Thai flood. In: 2012 Proceedings of PICMET '12: Technology Management for Emerging Technologies. pp. 2227–2232.

Le, X.-H., Ho, H.V., Lee, G., Jung, S., 2019. Application of long short-term memory (LSTM) neural network for flood forecasting. Water 11 (7), 1387.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L., 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL.

Li, W., 2020. GeoAI: Where machine learning and big data converge in GIScience. J. Spatial Inf. Sci. 20, 71–77.

Li, H., Caragea, D., Caragea, C., Herndon, N., 2017. Disaster response aided by tweet classification with a domain adaptation approach. J. Conting. Crisis Manag. 26 (1), 16–27. http://dx.doi.org/10.1111/1468-5973.12194.

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.

Mikolov, T., Chen, K., Corrado, G.S., Dean, J., 2013. Efficient estimation of word representations in vector space. In: ICLR.

Ming, X., Liang, Q., Xia, X., Li, D., Fowler, H.J., 2020. Real-time flood forecasting based on a high-performance 2-D hydrodynamic model and numerical weather predictions. Water Resour. Res. 56.

Mohammad, S.M., Turney, P.D., 2013. Crowdsourcing a word–emotion association Lexicon. Comput. Intell. 29.

Mojaddadi, H., Pradhan, B., Nampak, H., Ahmad, N., Ghazali, A.H., 2017. Ensemble machine-learning-based geospatial approach for flood risk assessment using multi-sensor remote-sensing data and GIS. Geomatics, Natural Hazards Risk 8, 1080–1102.

Moshe, Z., Metzger, A., Elidan, G., Kratzert, F., Nevo, S., El-Yaniv, R., 2020. HydroNets: Leveraging river structure for hydrologic modeling. arXiv:2007.00595.

Ning, H., Li, Z., Hodgson, M.E., Wang, C., 2020. Prototyping a social media flooding photo screening system based on deep learning. ISPRS Int. J. Geo-Inf. 9 (2), 104.

npm Inc, 2020. npm. Available online:https://npmjs.org/. (Accessed 18 October 2021).

OpenJS Foundation, 2020. Node.js. Available online: https://nodejs.org/. (Accessed 22 October 2020).

Pally, R., Samadi, S., 2022. Application of image processing and convolutional neural networks for flood image classification and semantic segmentation. Environ. Model. Softw. 148, 105285.

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.

Price, D.H., Hudson, K.L., Boyce, G., Schellekens, J., Moore, R.J., Clark, P.A., Harrison, T.G., Connolly, E., Pilling, C., 2012. Operational use of a grid-based model for flood forecasting.

Purver, M., Battersby, S.A., 2012. Experimenting with distant supervision for emotion classification. In: EACL.

Python Software Foundation, 2022. PyPI • the python package index. Available online: https://pypi.org/. (Accessed 16 May 2022).

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I., 2021. Learning transferable visual models from natural language supervision. In: ICML 2021: 38th International Conference on Machine Learning. pp. 8748–8763.

Ragini, J.R., Anand, P.R., Bhaskar, V., 2018. Big data analytics for disaster response and recovery through sentiment analysis. Int. J. Inf. Manage. 42, 13–24. http://dx.doi.org/10.1016/j.ijinfomgt.2018.05.004, URL https://www.sciencedirect.com/science/article/pii/S0268401217307843.

Riddell, H., Fenner, C., 2021. User-generated crisis communication: Exploring crisis frames on Twitter during hurricane harvey. Southern Commun. J. 86, 31–45.

Rout, J.K., Choo, K.-K.R., Dash, A.K., Bakshi, S., Jena, S.K., Williams, K.L., 2018. A model for sentiment and emotion analysis of unstructured social media text. Electron. Comme. Res. 18, 181–199.

Roux, H., Amengual, A., Romero, R., Bladé, E., Sanz-Ramos, M., 2020. Evaluation of two hydrometeorological ensemble strategies for flash-flood forecasting over a catchment of the eastern pyrenees. Nat. Hazards Earth Syst. Sci. 20, 425–450.

Sahni, T., Chandak, C., Chedeti, N.R., Singh, M., 2017. Efficient Twitter sentiment classification using subjective distant supervision. In: 2017 9th International Conference on Communication Systems and Networks. COMSNETS, pp. 548–553.

Said, N., Ahmad, K., Gul, A., Ahmad, N., Al-Fuqaha, A.I., 2020. Floods detection in Twitter text and images. MediaEval.

Sakaki, T., Okazaki, M., Matsuo, Y., 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In: The Web Conference.

Schoene, A.M., Dethlefs, N., 2016. Automatic identification of suicide notes from linguistic and sentiment features. In: LaTeCH@ACL.

Sloan, L., Morgan, J., 2015. Who tweets with their location? Understanding the relationship between demographic characteristics and the use of geoservices and geotagging on Twitter. PLoS One 10 (11), e0142209.

Smith, L., Liang, Q., James, P., Lin, W., 2017. Assessing the utility of social media as a data source for flood risk management using a real-time modelling framework. J. Flood Risk Manag. 10 (3), 370–380.

Teng, J., Jakeman, A., Vaze, J., Croke, B., Dutta, D., Kim, S., 2017. Flood inundation modelling. Environ. Model. Softw. 90 (90), 201–216.

TensorFlow Contributors, 2019. Tensorflow. Available online: https://www.tensorflow.org/. (Accessed 06 January 2020).

Vashishtha, S., Susan, S., 2019. Fuzzy rule based unsupervised sentiment analysis from social media posts. Expert Syst. Appl. 138, 112834.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 5998–6008.

Vichiantong, S., Pongsanguansin, T., Maleewong, M., 2019. Flood simulation by a well-balanced finite volume method in tapi river basin, thailand, 2017. Model. Simul. Eng..

de Vitry, M.M., Kramer, S., Wegner, J.D., Leitão, J.P., 2019. Scalable flood level trend monitoring with surveillance cameras using a deep convolutional neural network. Hydrol. Earth Syst. Sci. 23 (11), 4621–4634.

Wang, R.-Q., Mao, H., Wang, Y., Rae, C., Shaw, W., 2018. Hyper-resolution monitoring of urban flooding with social media and crowdsourcing data. Comput. Geosci. 111, 139–147. http://dx.doi.org/10.1016/j.cageo.2017.11.008.

Wu, W., Emerton, R.E., Duan, Q., Wood, A.W., Wetterhall, F., Robertson, D.E., 2020. Ensemble flood forecasting: Current status and future opportunities. Wiley Interdiscipl. Rev. Water 7.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V., 2019. XLNet: Generalized autoregressive pretraining for language understanding. In: NeurIPS.

Yin, J., Lampert, A., Cameron, M., Robinson, B., Power, R., 2012. Using social media to enhance emergency situation awareness. IEEE Intell. Syst. 27 (6), 52–59. http://dx.doi.org/10.1109/mis.2012.6.

Zhang, T., Xu, B., Thung, F., Haryono, S.A., Lo, D., Jiang, L., 2020. Sentiment analysis for software engineering: How far can pre-trained transformer models go? In: 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME). pp. 70–80.