remote sensing

MDPI

Article

Measuring the Impact of Natural Hazards with Citizen Science: The Case of Flooded Area Estimation Using Twitter

Pierrick Bruneau [1,*], Etienne Brangbour [1,2], Stéphane Marchand-Maillet [2], Renaud Hostache [1], Marco Chini [1], Ramona-Maria Pelich [1], Patrick Matgen [1] and Thomas Tamisier [1]

[1] Luxembourg Institute of Science and Technology, L-4361 Esch-sur-Alzette, Luxembourg; etienne.brangbour@list.lu (E.B.); renaud.hostache@list.lu (R.H.); marco.chini@list.lu (M.C.); ramona.pelich@list.lu (R.-M.P.); patrick.matgen@list.lu (P.M.); thomas.tamisier@list.lu (T.T.)

[2] Computer Science Department, University of Geneva, 1227 Carouge, Switzerland; stephane.marchand-maillet@unige.ch

* Correspondence: pierrick.bruneau@list.lu

Abstract: Twitter has significant potential as a source of Volunteered Geographic Information (VGI), as its content is updated at high frequency, with high availability thanks to dedicated interfaces. However, the diversity of content types and the low average accuracy of geographic information attached to individual tweets remain obstacles in this context. The contributions in this paper relate to the general goal of extracting actionable information regarding the impact of natural hazards on a specific region from social platforms, such as Twitter. Specifically, our contributions describe the construction of a model classifying whether given spatio-temporal coordinates, materialized by raster cells in a remote sensing context, lie in a flooded area. For training, remotely sensed data are used as the target variable, and the input covariates are built on the sole basis of textual and spatial data extracted from a Twitter corpus. Our contributions enable the use of trained models for arbitrary new Twitter corpora collected for the same region, but at different times, allowing for the construction of a flooded area measurement proxy available at a higher temporal frequency. Experimental validation uses true data that were collected during Hurricane Harvey, which caused significant flooding in the Houston urban area between mid-August and mid-September 2017. Our experimental section compares several spatial information extraction methods, as well as various textual representation and aggregation techniques, which were applied to the collected Twitter data. The best configuration yields a F1 score of 0.425, boosted to 0.834 if restricted to the 10% most confident predictions.

Keywords: social media; flooded area estimation; classification; citizen science; volunteered geographic information

Citation: Bruneau, P.; Brangbour, E.; Marchand-Maillet, S.; Hostache, R.; Chini, M.; Matgen, P.; Tamisier, T. Measuring the Impact of Natural Hazards with Citizen Science: The Case of Flooded Area Estimation Using Twitter. *Remote Sens.* **2021**, *13*, 1153. https://doi.org/10.3390/rs13061153

Academic Editor: Tal Svoray

Received: 9 February 2021
Accepted: 11 March 2021
Published: 18 March 2021

1. Introduction

Several authors have explored the use of Twitter in the context of environmental hazards prevention and mitigation in the literature. For example, Twitter is used by Sakaki et al. [1] to help damage detection and reporting in the context of Earthquake events. The TAGGS platform aims at using Twitter for flood impact assessment at a global scale [2]. Twitter has also been used to monitor the spread of the seasonal flu disease [3]. Jongman et al. [4] use Twitter in order to trigger humanitarian actions for early flood response. Specifically, the authors mainly use Twitter to raise an alert over a region, which is coupled to the Global Flood Detection System (GFDS) [5] in order to trigger near real-time mapping, which is in a way that keeps up with GFDS data update frequency. Wiegmann et al. [6] identify the prime promise of filling in information gaps with citizen observations, issued by *citizen sensors*. Opportunities for social media are identified as helping impact assessment and model verification, and for strengthening the acquisition of relevant information. Their literature review concludes that social media content is unlikely to increase spatial resolution of traditional sources, such as remotely

sensed data, but it bears potential for increasing the temporal resolution. These conclusions hold promise for social media in flood management, as several authors [7–9] identified temporal revisit as one of the main limitations of satellite imagery in flood monitoring and forecasting. Many studies [1,2,10–12] that address the use of Twitter for improving the response to natural disasters mainly focus on analyzing the spatial dimension of a Twitter corpus (i.e., a set of Twitter posts). Geographic information is present in tweets in the form of discrete GPS coordinates (often referred to as *geotags*) and surface bounding boxes representing toponyms. Among these, geotags are the most accurate, hence the most interesting *a priori*. However, several studies report that only about 1% of all tweets are holding a geotag [13–17]. In response to this lack of geographic information, several authors have focused on the means to further localize Twitter content [2,18].

Named Entity Recognition (NER) aims at parsing entities in text, for example, people or toponyms. We focus on the latter in the context of this paper. Twitter text differs significantly from typical textual content, such as news articles from several perspectives that hamper the ability of NER systems to effectively extract entities. In particular, tweets are very short, and they are often not grammatically or syntactically correct, in contrast to more verbose and curated inputs that are generally expected by textual analysis algorithms. The adaptation of NER systems to the peculiarities of Twitter content is still an active field of research [19]. For example, we pinpoint the TwitterNER system [20], for which the source code and training data for the English language have been released (https://github.com/napsternxg/TwitterNER (accessed on 17 March 2021)). In short, this system combines unsupervised word representations to gazeteers, such as GeoNames and string matching. It embeds a tokenizer that is specially designed for Twitter in English [21]. De Bruijn et al. [2] use NER adapted to Twitter to extract geographic entities that are then combined into a resolution index table that also exploits tweet grouping. They address the consistency of multiple cues at a large geographic scale. In a similar approach, Schulz et al. [18] combine geotags, NER results, bounding boxes, user information, and emission time zones in a polygon stacking approach.

Grace [17] claims that one-third to half of disaster-related content in social media features toponyms. He distinguishes coarse-grained (regional) from fine-grained (local) geographic information in Twitter, and finds that coarse-grained information is more common. In support of this claim, in a recent technical report [22] on the Twitter corpus described in Section 5.2, we found that only 3.57% of toponym mentions are associated to surface bounding boxes that are smaller than 350 km^2. Consequently, Twitter is less suitable for fine-grained situational awareness *a priori*. Geotags also suffer from a discrepancy. For example, in automated storm-related reports, coordinates sometimes lie outside of the zone referred to in the text [17]. This is coined as the *sensor-subject displacement* problem [23].

The baseline way of predicting whether a point in space and time will be flooded is to simulate the water flow and runoff that result from the expected rainfall. For instance, Lisflood-FP forecasts water heights reached in a region using such a physical model fed by ground measurements [24]. Data assimilation is a common way to incorporate in-situ or remote sensing derived observations into such forecasting models with the aim of improving their predictions [7,9,25,26].

Brouwer et al. [27] compute flood probability maps from Twitter, and they mitigate the spatial sparsity of its geographic information with the Height-Above-Nearest-Drainage (HAND) model [28]. This model lets them infer areas implicitly flooded if a given reference point, as reported in a tweet, is assumed to be flooded. In practice, their procedure expands the area of influence of a set of manually curated tweets as much as possible, with a view to maximizing the predicted area.

The HAND model boosts the utility of a Digital Elevation Model (DEM) by accounting for the local river elevation [29]. Alternatively to the HAND model, Eilander et al. [30] account for a DEM in an advanced way by computing local depressions while using the PCRaster software [31]. A flood-fill algorithm assuming flat horizontal water surface is applied to model results. Their forecasts are validated using geo-located photos.

Fohringer et al. [32] use a DEM in a similar manner, but with a view to deriving water levels over a region from water depths manually estimated from photographs.

In order to facilitate textual content processing in a general manner, and Twitter messages in particular, a necessary step is to convert textual items to a numerical representation (often referred to as *feature vectors* [20,33,34], or *embeddings* [35,36] in the machine learning literature). A classification model may then learn the relationship between these representations and a target concept, for example, *does the message reveal a flood event?* in the context of this paper. The most immediate representation is the binary value indicating whether the text string matches any of pre-defined keywords, known as the *keyword-matching* feature in the literature [37,38]. *TFIDF* (Term-Frequency-Inverse-Document-Frequency) vectors [33] are also often used in order to represent textual documents. Dimensions in this feature vector are attached to the vocabulary of all possible phrases, and their non-negative values are given by the product between the frequency of respective phrases in the document and their inverse frequency in the whole document corpus. Intuitively, the value of a dimension is greater as the respective phrase characterizes the document, in other words because it is frequent in the document and rare in the remainder of the corpus. In the context of *TFIDF*, phrases are tuples of n words, often known as *n-grams*. The simpler option is to only use single words (one-grams), but the union of one-grams and two-grams has often been considered in the literature [34,37,38]. A simpler, binary version of *TFIDF*, which assigns 1 to the respective dimension when a given phrase is found in the text, and 0 else, has also been witnessed [39].

The dimensionality of *TFIDF* feature vectors is determined by the size of the vocabulary, which can be quite large. Additionally,the *TFIDF* feature vectors of Twitter messages are very sparse, which is problematic for most statistical learning models. Neural language models were recently introduced as a mean to generate numerical embedding vectors. They are generally trained on large corpora offline, and used on smaller test collections. The obtained feature vectors are more compact and denser (generally a few hundred dimensions) at the cost of interpretability, which is individual dimensions cannot be attached to semantics, unlike *TFIDF*. A well-known neural language model is *Word2Vec* [40], which is trained unsupervisedly and generatively to map textual documents to feature vectors. However, similarly to *TFIDF*, this model is phrase-based, and it is trained on well-curated text, which is unadapted to Twitter content. *Tweet2Vec* circumvents these issues by considering a character-based model [35]. It is trained self-supervisedly on a hashtag prediction task. Additionally, instead of a shallow neural network in *Word2Vec*, the authors use a bidirectional LSTM model [41].

Keyword matching is implemented by the Twitter API, and it has served to collect a corpus related to Hurricane Harvey with keywords *Hurricane Harvey*, *#HurricaneHarvey*, *#Harvey*, and *#Hurricane* during the event [42]. In the context of a flu spread analysis, Gao et al. [39] also isolated a corpus using this technique. However it is likely to inaccurately reflect whether a tweet is *semantically* related to a flood event. To circumvent this issue, Gao et al. manually annotated 6500 tweets in order to train a SVM classifier that further filters out false positives. In its counterpart, this approach requires heavy manual annotation work. Instead, when a spatial distribution is available for the target event (e.g., rainfall [34] or flood [27]), an alternative approach is to aggregate tweet feature vectors with respect to spatial cells, and then learn a function that maps the aggregated representation to the target values. In the context of data assimilation mentioned above, this kind of approach opens the possibility to learn a mapping function from a corpus of tweets that were collected at time t to a related ground truth map (e.g., remotely sensed data at time t) and, hence, increase the temporal frequency of proxy observations by using a new test corpus of tweets that were collected at time $t + \delta t$.

2. Author Contributions and Position

Volunteered Geographic Information (VGI) is the umbrella domain that denotes the contribution of geographic information by the public in a participatory fashion, and the

processing and exploitation of this information [2,13–15]. This domain encompasses contributions to the OpenStreetMap database, as well as people geotagging photos showing flood extents on social networks. In particular, Craglia et al. [13] distinguish explicit and implicit VGI, depending mostly on the initial purpose of information contributions. The contribution to OpenStreetMap is a typical example of explicit VGI: the contributor purposefully updates the database. On the other hand, the initial aim of a Twitter user witnessing a natural hazard, such as a storm or a flood, and attaching geographic information, such as a geotag or a toponym mention to her message, is to inform her followers about what she witnesses.

In the context of natural hazards, VGI is presented as an interesting way to provide data that are useful for emergency response personnel, emergency reporting, civil protection authorities, and the general public [2,13]. It comes in complement to more traditional information sources, such as forecasts obtained from hydrological and hydraulic models [7], and as an additional tool for situational awareness. Additionally, social platforms, such as Twitter, are constantly updated, which opens the possibility to picture the situation in impacted regions quickly and regularly.

Our technical contributions relate to the exploitation of implicit VGI, which poses specific challenges. First, the event sought is intertwined with many topics in the Twitter stream of posts. The identification of relevant implicit VGI is then not trivial. Second, in contrast to standard GIS formats, geographic information in Twitter is provided in several heterogeneous forms (geotags, spatial bounding boxes, user information, and toponym mentions in the tweet text) with various levels of accuracy and reliability.

In this work, we formalize mapping function fitting as the classification task parametrized by a coordinate space. This model allows for us to implicitly account for the spatio-temporal dimensions in a classical statistical learning context. We also formalize the way a set of tweets collected in a given time frame parametrizes a feature vector mapping function, which associates aggregated feature vectors to coordinate points. We show how to compute a feature vector that reflects the local diversity of aggregated tweets.

For the experiments, we focus on flooded area estimation, and make use of a new corpus of tweets that were collected during Hurricane Harvey, which has affected the Houston urban region between mid-August and mid-September 2017. We presented this corpus at length in a dedicated technical report [22], recalled in Section 5.2. We compare several textual representations and spatial extraction methods, which affect the obtained feature vector mapping function. We address concerns regarding Twitter spatial information accuracy [23], by issuing recommendations for filtering the corpus at hand, and quantified consequences in terms of imbalance and support, contributing to addressing the challenges that are posed by implicit VGI.

The present work is heavily inspired by Lampos and Cristianini [34], and it is incremental with respect to one of our previous contributions [43]. From the former, we took the idea of mapping aggregated vectors to a target spatial variable, and then moved it to the context of a much finer-grained spatial context (Lampos and Cristianini [34] perform the classification task at a city neighborhood scale), in order to extend the range of possibilities offered by VGI. We improved the latter with a more general formalism in Section 3, and a more comprehensive address of VGI through the combination of Twitter geographic information, a DEM, and toponyms extracted using NER techniques. We also compared several solutions for feature vector construction, involving advanced machine learning techniques, such as neural embedding models [35] and Fisher vectors [44]. Spatial information extraction methods, combining one or more sources among place bounding boxes, geotags, and toponyms obtained by NER techniques, are compared with respect to F1, precision, and recall scores. Finally, we describe a qualititative comparison between predictions that were achieved by the proposed method, and labels obtained by crowdsourcing.

The objective of this study is significantly different from Brouwer et al.'s (2017) related work on flood extent prediction [27]. In their study, the authors attempt to infer flood boundaries from a small amount of carefully validated tweets, whereas, here, we aim to

identify a subset of raster cells with high confidence in their classification as a flooded or dry area. To this end, we make use of a larger amount of automatically collected tweets, with minimal manual effort.

Here, we aggregate all tweets, irrespective of their relevance to the subject, and learn feature combinations that enable the prediction of the ground truth class. The spatial uncertainty of Twitter content is accounted for by having the parameters of an aggregation function varying with the bounding box size, giving higher weight to content with more precise spatial information.

Specifically, in this work, we retain all tweets that match the requirements in featuring geographic information accuracy (as discussed in Section 5.4), and learn decisive features without traditional supervision at the tweet level. Here, the supervision comes from a ground truth on geographic raster cells being flooded or not, and useful features are aggregated at a spatial cell level. This design can be seen as a distant supervision variant [45], where spatial aggregation acts as a bridge to connect labels to textual representation features. Unrelated content is implicitly discarded by the model building procedure that is presented in Section 3, which learns to exclude irrelevant features from aggregated vectors.

The authors of the present study have contributed related techniques for extracting flood extent observations from SAR imagery using a hierarchical split-based approach [46], which was recently improved in urban areas using temporal coherence between SAR acquisitions [47,48]. The target variable in our experiments, as presented in Section 5.1, was actually generated using one of these techniques. Besides advancing VGI in general, a motivation for the present work is to show that a mapping between information extracted from Twitter and SAR imagery can be built, hence opening the computation of a flood observation proxy at a much higher temporal frequency than what is currently possible with remote sensing derived information. This perspective is highly promising for improving flood forecasting using techniques from the data assimilation domain, to which the authors of this study have also contributed [9].

3. Spatial Mapping Problem Definition

In this section, we formalize a classification problem, which relates feature vectors to a spatially distributed binary target variable (*flooded or not flooded*). Let us consider a coordinate space Δ. In the context of remotely sensed data, points $p \in \Delta$ generally lie on a regular spatio-temporal grid. We also consider the binary target mapping function $y : \Delta \to \{0, 1\}$, with $y(p) = 1$ if the spatio-temporal cell p is flooded. We assume that a multivariate feature mapping function $x : \Delta \to \mathbb{R}^d$ is also available, and it is linked to the binary function by another function $f : \mathbb{R}^d \to \{0, 1\}$ so that $y(p) = f(x(p))$. In practice, training feature vector and target couples $\{x(p), y(p)\}_{p \in \Delta_{\text{train}}}$ are provided, and the best possible function f^* is found, which minimizes the cross-entropy loss:

$$f^* = \arg_f \min -\frac{1}{|\Delta_{\text{train}}|} \sum_{p \in \Delta_{\text{train}}} y(p) \ln f(x(p)) + (1 - y(p)) \ln(1 - f(x(p))) \quad (1)$$

The trained model can then be used to predict the target variable y for coordinates $p \in \Delta_{\text{test}}$ unseen at training time. In the remainder of the paper, without a loss of generality, we assume that training data are collected for a single point in time, which is all training coordinates lie in a spatial grid for the same given day. The trained model would then use coordinates for other days as test data. This design choice conforms to the alleged objective of providing a flooded area proxy estimator with higher temporal frequency, as expected in the literature [6]. Hence, training coordinates refer to a single day: for simplicity in the remainder of the paper, coordinate points in equations will leave the temporal dimension implicit, and refer to the *(longitude, latitude)* couple.

The loss function in Equation (1) is minimized by models, such as the logistic regression model and classification neural networks, while using nonlinear optimization [49] or gradient descent algorithms [50], depending on the model at hand. Fitting a classification model with respect to a coordinate space has already been considered in the

literature [34,39,51]. However, the formalism here is more general, as we are not tied to a specific model architecture, as long as it minimizes the required loss function. The computation of the multivariate feature vector mapping function is the crux of our methodology. This aspect is discussed in the next section.

4. Feature Vector Mapping Function

We now detail the construction of the feature vectors (left implicit in the previous section) by weighted aggregation, and expose several aggregation variants. Let us consider a collection of tweets $\mathcal{C}_N$:

$$\mathcal{C}_N = \{p_n, d_n, x_n\}_{n \in 1...N} \tag{2}$$

with $p_n \in \Delta$ the spatial coordinates of the tweet, d_n its spatial dispersion, and x_n its feature vector representation, which we assume as homogeneous to the multivariate feature vectors $x(p)$ in Equation (1). Numerical vector representations would typically be obtained using language models, such as *TFIDF* [33] and *Word2Vec* [40]. Confirmed representations are presented in Section 5.3. The dispersion reflects the geographic precision of the tweet. In this paper, we use the surface of a geographic bounding box as dispersion. The feature mapping function for coordinates $p \in \Delta$ is generated from $\mathcal{C}_N$, as:

$$x(p) = \sum_{n=1}^{N} w_n(p, p_n, d_n) x_n \tag{3}$$

with weight w_n obtained from:

$$w_n(p, p_n, d_n) = \mathcal{N}(p; p_n, d_n \mathbf{I}) \tag{4}$$

The weight w_n for spatial coordinates p with respect to the nth tweet is hence obtained from the 2D Gaussian distribution $\mathcal{N}$ centered on the tweet spatial coordinates p_n with variance d_n. Hence, the importance of the nth tweet in the feature vector for coordinates p will be greater as the tweet coordinates get close to p, and its dispersion is small (i.e., tweet geographic information is accurate). Let us note that, if the temporal coordinate was to be included in the coordinate space, specific mean and dispersion terms should be added to Equation (4).

Gao et al. [39] used the Epanechnikov kernel function instead of the Gaussian distribution in Equation (4), in combination to the *keyword-matching* binary feature. Lampos and Cristianini [34] discretized the coordinate space Δ into neighborhoods, and used constant weights when computing per-neighborhood feature vectors with Equation (3). The weighting function in Equation (4) is also closely related to the Inverse Distance Weighting interpolation described Brouwer et al. [27], which involves a heavy tailed improper distribution instead.

In this paper, we consider additional variants to weight and feature vector computation. First, we propose to account for topographical features with an alternative weight computation function:

$$w_n^t(p, p_n, d_n) = w_n(p, p_n, d_n) I(p, p_n) \tag{5}$$

where the t superscript stands for *terrain*. The values for I are obtained by applying a flood-fill algorithm (https://scikit-image.org/docs/dev/auto_examples/segmentation/plot_floodfill.html (accessed on 17 March 2021)) originating from the tweet coordinates p_n. The flood-fill algorithm is parametrized by a DEM. At this stage, we assume the availability of this DEM and that its resolution matches the grid on which coordinates p_n lie: implementation details regarding the DEM are disclosed in Section 5.1. Subsequently, under the hypothesis that p_n is flooded according to the ground truth y, $I(p, p_n)$ is 1 if p is flooded as a consequence of the flood-fill algorithm, and 0 otherwise. In practice, flood-filling introduces a spatial regularization to w, which favors the propagation of tweet information in a consistent way regarding the DEM. Information extracted from a

DEM was also accounted for in similar contexts. Fohringer et al. use a DEM to propagate water levels that were estimated visually in photographs with known locations [32]. Water heights are alternatively identified in the text by Eilander et al. [30]. However, these two approaches proceed by propagating information from a low number (less that 100) of carefully curated tweets, whereas our approach is fully automated, and accounts for a much larger (a few thousands in our experiments), and possibly more noisy, amount of tweets. Brouwer et al. [27] proposed a more advanced method using the HAND model. However, we argue that the flood-fill algorithm is sufficient for our purpose to mask weights that characterize a small area around a given tweet.

We also considered the adaptation of Fisher vectors [44] to our feature vector mapping context. We first proceed by extracting K clusters from a collection of feature vectors, in our case $\{x_n\}_{n\in 1...N}$. Clusters are obtained from a Gaussian Mixture Model, which is estimated using the EM algorithm [52]. The model outputs the membership probabilities of feature vectors a_n, so that a_{nk} is the probability that x_n belongs to cluster k. Krapac et al. [44] then compute the gradient of the log-likelihood of the set of feature vectors that were extracted from a test image with respect to the clustering model parameters, namely the cluster weights, means, and covariance matrices.

In this paper, we limit ourselves to taking gradients with respect to cluster weights θ_k in order to limit the dimensionality of the resulting Fisher vector ($K = 200$, as recommended by Krapac et al. [44]). Additionally, instead of using an independent test image, we consider the feature vectors in $\mathcal{C}_N$, weighted with respect to the coordinates p (Equation (4)). Krapac et al. show that the gradient of the log-likelihood of x_n with respect to θ_k is $a_{nk} - \theta_k$. Translating to our case, the k^{th} dimension of the Fisher vector results in:

$$x_k^F(p) = \frac{\sum_n^N w_n(p, p_n, d_n)[a_{nk} - \theta_k]}{\sum_n^N w_n(p, p_n, d_n)} = \frac{\sum_n^N w_n(p, p_n, d_n)a_{nk}}{\sum_n^N w_n(p, p_n, d_n)} - \theta_k \quad (6)$$

Intuitively, the Fisher vector of a raster cell is a multi-dimensional measure of the extent to which the collection of tweets specific to this cell diverges from the overall tweet distribution.

5. Data and Preprocessing

In this section, we describe the map data feeding the target variable in the problem formulation that is given in Section 3. We also disclose the criteria that directed the collection of our Twitter data corpus. Subsequently, we enumerate the textual representation models that were tested in this paper. We recall that they allow associating numerical feature vectors to pieces of text, hence building x_n in Equation (2). Afterwards, we detail how the relevant spatial information was extracted from the Twitter corpus, and enriched with NER from Twitter text.

5.1. Target Map Construction

As the ground truth for our learning procedure ($y(p)$ in Equation (1)), we consider the flood map that is shown in Figure 1a. Hurricane Harvey affected the Houston urban region between mid-August and mid-September 2017, with a flooding peak around the 30 August 2017. The map originates from Sentinel-1 SAR images that were acquired over the metropolitan area of Houston both during the flood event on 30 August 2017 as well as prior to the flood event on 18 and 24 August 2017. The Sentinel-1 mission is a constellation of two SAR satellites of the European Copernicus programme. The SAR images were transformed using a method that was contributed by authors of this paper [47]. Because of the requirements of the present work, it was rescaled to 2.10^{-3° resolution (approx. 200 m). The map features approximately 1.23 M pixels, among which 3.1% are reporting flooded areas. After masking permanent waters from this map, 840 k pixels remain, among which 4.4% are marked as flooded, thus allowing for a slight rebalance of the target data set. In the end, for coordinates p falling in its non-masked area, the map returns a

binary value indicating whether the associated area is flooded or not according to remote sensing sources.

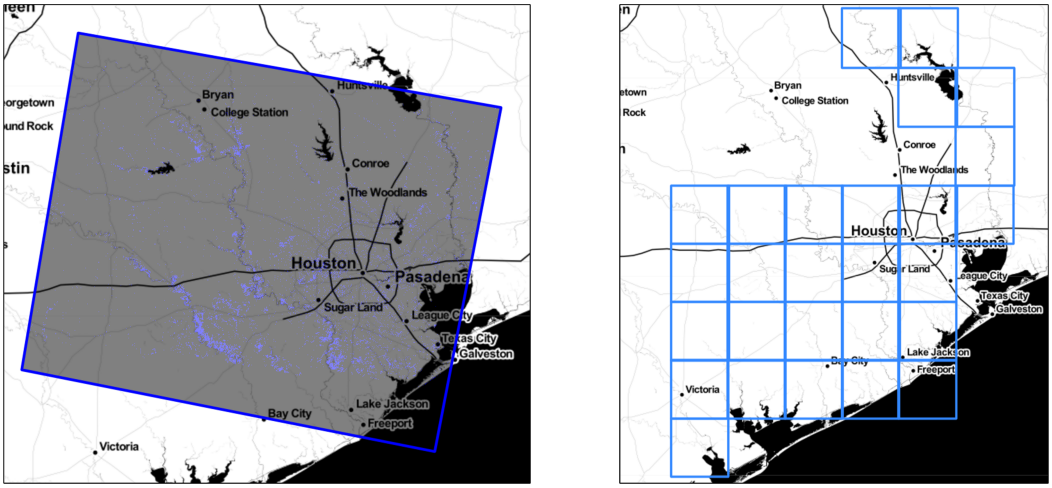

Figure 1. Flood map derived from Sentinel-1 SAR images (**a**). Spatial bounds for the query used to build our Twitter corpus related to Harvey (**b**). The map tiles are by Stamen Design.

The DEM for the region enclosing the spatial bounds in Figure 1b, which parametrizes Equation (5), was obtained from the US Geological Survey (https://viewer.nationalmap.gov/basic (accessed on 17 March 2021)). The obtained raster file has 10^{-4° resolution. We converted it to the same coordinate system as the SAR-derived data that are described above, and rescaled it with respect to the nearest neighboring pixels, so that its resolution equals 2.10^{-3°.

5.2. Twitter Data Collection

A corpus of tweets collected during the Harvey Hurricane has been made available shortly after the event [42]. It features tweets matching any of the phrases *Hurricane Harvey*, *#HurricaneHarvey*, *#Harvey*, and *#Hurricane*. Using the Twitter interfaces, we also collected our own corpus of 7.5 M tweets that were related to the event. Tweets are obtained as *JavaScript Object Notation* (JSON) items. Under this format, each tweet is a set of key-value pairs. Among the variety of meta-data distributed by Twitter, we identified *coordinates* (the geotag), *place.full_name* (the toponym), *place.bounding_box* (the spatial bounding rectangle that is associated to the toponym), and *text* (the actual tweet) as the pieces of information relevant to our work.

In order to match the scope and objectives disclosed in the introduction, especially regarding content localization, we did not use textual query filters, and collected all tweets with either the attached bounding box or geotag overlapping the spatial bounds of the Houston urban surroundings between the 19 August and the 21 September 2017. The spatial area of interest, as shown in Figure 1b, has been determined according to posterior analyses of Hurricane Harvey impacts. We found a very significant positive correlation ($p < 10^{-10}$ at Pearson's and Kendall's tests) between toponym frequency in the corpus and the associated bounding box surface. Such correlation is quite natural, as toponyms, such as *Houston*, are indeed much more likely to be attached as meta-data to tweets than very specific places, such as *Cypress Park High School*. As a consequence, only a small portion of a Twitter corpus collected according to geographic criteria will be actually usable for fine-grained purposes, such as identifying flooding areas at a city scale. Figures that are obtained in the context of the requirements of the present work, as given in Section 5.4, confirm this conjecture. With this regression analysis, we also qualitatively identified the surface of 350 km^2 as the threshold delineating large (city-scale) toponyms,

from local-scale toponyms, such as streets or malls. We provide additional details regarding the corpus collection, pre-processing, and descriptive analysis in a dedicated technical report [22].

5.3. Textual Representation

The Twitter corpus definition that we use in Equation (2) assumes that individual tweets can be represented by a numerical vector, but it leaves the transformation method actually used implicit. The act of numerical embedding, which is transforming a piece of text into a numerical vector, has been covered in the introduction. In this experimental section, we compare three embedding methods from the literature. For each method, we indicate specific implementation and pre-processing details.

The *keyword-matching* feature equals 1 if the tweet text matches any of predefined keywords, after converting it to lowercase. Therefore, the resulting feature vector has a single dimension. We use *harvey* and *flood* as keywords reflecting the relatedness to the flooding event, as we showed these keywords are more effective than those used by Littman [42] in a previous contribution (see Section 2).

Given a vocabulary of V phrases, the *TFIDF* model [33] is defined as:

$$\begin{aligned} \mathrm{tf}_n(v) &= \text{ times } v \text{ is matched in tweet } n \\ \mathrm{idf}_n(v) &= \ln(1+N) - \ln(1+N_v) + 1 \\ x_n^{TFIDF} &= \{\mathrm{tf}_n(v) \times \mathrm{idf}_n(v)\}_{v \in V} \end{aligned} \tag{7}$$

with $N_v = |\{n' \in 1 \dots N | \mathrm{tf}_{n'}(v) > 0\}|$. Notations in Equation (7) refer to the Twitter corpus definition given in Equation (2). For *TFIDF*, as suggested in Lampos and Cristianini [34], we adapt Twitter-specific features. Specifically, user references (for example, *@Bob*) are removed, hashtags are expanded as regular text (accounting for camel case, for instance *#HarveyFlood* becomes *Harvey Flood*). We also filter out URLs. Punctuation is removed, and Porter stemming [53] is applied to the result. This pre-processing is similar to that presented by Dittrich [19] in the context of NER. Following the recommendations by Lampos and Cristianini [34], we consider the set of one and two-grams as the dimensions of the *TFIDF* space. Simply put, dimensions in *TFIDF* are associated to single words in the one-gram model, whereas two-word phrases are also included in the two-gram model. *TFIDF* yields a very high-dimensional space (6618 with one-grams alone, 26,125 with one and two-grams, averaged over experimental conditions in this section). Actually, many of these tokens are present only a few times in $\mathcal{C}_N$: as suggested by Lampos and Cristianini [34], we retain features that appear at least 10 times in $\mathcal{C}_N$. This parametrization yields 534 one-grams, and 188 two-grams, hence the dimensionality of *TFIDF* space is 722 (averaged over experimental conditions reported in this section). The obtained feature vectors are normalized to unit norm.

As the neural language embedding method, we consider the *Tweet2Vec* model that was proposed by Dhingra et al. [35]. It uses a bidirectional LSTM [41]. The model is character-based in order to mitigate Twitter-specific language artifacts (e.g., poor syntax, abbreviations). It updates a latent state vector using bidirectional passes on the sequence of tweet characters. Hashtags are removed from tweets beforehand, and the model is trained to predict them. Dhingra et al. [35] showed that the logistic regression layer that relates the eventual latent vector to hashtag classes captures some of the corpus semantics. The model is trained as specified in the reference paper. The size of the output embedding is the only notable difference in our implementation, which is set to 500 in the *Tweet2Vec* paper. Experimentally, we found that it could be reduced to 200 with negligible consequences in terms of performance (less than 1% reduction of the macro-averaged F1 score, see Section 6.2 for the definition of the F1 score). The models were trained using a corpus of 1.3 M tweets that were sampled from all tweets sent in English during one week. URLs are removed, and user references are kept after replacing all user references by the same *@user* placeholder. In test conditions with our Harvey corpus, similarly to the *TFIDF* case,

the text in hashtags is also kept alone, just removing the hash key and expanding the camel case text. Dhingra et al. [35] completely remove the hashtags, as they are the target classes to be predicted for training the language embedding vectors. We keep this text when the point is to compute a representative embedding vector, as the text behind the hash key may also have semantic value.

5.4. Spatial Information Extraction

In the context of this paper, we focus on tweets sent on the 30 August. For this day, 310 k tweets are stored in our database. As already discussed in the introduction, two main pieces of geographic information are optionally attached to tweets: geotags and place bounding boxes. Among tweets sent on the 30 August, 4501 (ca. 1.5%) are holding a geotag. This proportion is close to expectations according to the literature [14]. The range of surfaces associated to place bounding boxes is very large, from the street to city scale. As the target application in this paper is fine-grained, we considered only bounding boxes with at most 20 km^2 (1.6×10^{-3} deg^2) associated surface. On the 30 August, 2630 tweets (ca. 0.8%) match this specification. We note that only 244 tweets fall in the overlap between selections based on place or geotag, so these two collections are complementary. As the utility of geotags for a fairly high resolution target has been questioned in the literature [23,54–56], in our experiments we will test all of the alternatives (bounding box alone, geotag alone, joint use). We note that alternatives cause a variation of the size of $\mathcal{C}_N$ (Equation (2)), which serves as basis for the computation of the feature vectors ($N = 2630$ in the former case, 6887 in the latter case).

The surface of geographic bounding boxes (in squared degrees) is used as the dispersion parameter, which drives the feature vector mapping function, as mentioned in Section 4. This parameter may be zero if the only geographic information attached to the tweet is a geotag, or if the bounding box has zero length or width (occurs in the collected data). Such a case would cause a degenerate optimization problem in Equation (1), as infinite weights are then issued from Equation (4). To prevent this problem, we lower bound d_n by $d_{\min} = 8 \times 10^{-5}$ deg^2. This threshold was set, so that the weight at the center of this minimal bounding box is 20 times the weight at the center of a 20 km^2 bounding box. When both the geotag and place bounding box are attached to a given tweet, we use $d_{\min}$ as its dispersion, in order to reflect that the geotag is the most accurate *a priori*. Additionally, in this case, we use the geotag as p_n, which is when both the place bounding box and geotag are present, we ignore the spatial information that is carried by the bounding box. It is worth noting that the usable data are only a small portion of the total amount of collected tweets. However, the quantity of tweets used is still two orders of magnitude larger than that witnessed in related work, such as Fohringer et al. [32] and Eilander et al. [30].

5.5. Named Entity Recognition

We used the TwitterNER system to augment the spatial information described in the previous section [20]. Extracted toponyms were geocoded using a custom Nominatim (https://github.com/mediagis/nominatim-docker (accessed on 17 March 2021)) instance that was restricted to the smallest available database file enclosing the region of interest in the experiments (Texas). Specifically, we retain extracted location entities, and retrieve associated candidate place bounding boxes from the Nominatim instance. Nominatim searches may return multiple results. We chose to exclude identified NER locations with the first returned result associated to a surface larger than 350 km^2 (see Section 5.2 for the justification of this threshold) in order to ignore place mentions, such as *Houston*, which are not useful regarding the accuracy requirements in this paper. Multiple returned results are stored after checking for the overlap with the area of interest highlighted in Figure 1b. Entities having their first matching results not overlapping the area of interest in the study were also excluded. Thus, over the whole corpus, 72,026 location named entities were extracted, which were associated to 36,715 distinct tweets. These mentions refer to 5249 unique location names.

Because Twitter is a noisy text source, and TwitterNER may output false positives, the extracted location names feature obvious mismatches (for example, Cali, short for *California*, but mistaken for *Cali Drive* in the results returned by Nominatim). Hence, we manually curated a subset of the identified location names. Empirically, we found that the sorted frequencies of extracted location names are exponentially decreasing: therefore, the curation effort was focused on the 1000 most frequent location names, which account for 90% of the extracted mentions. In practice, we curated 1175 location names, among which 711 were identified as valid and 444 marked as mismatches. The curation resulted in 38 k valid and 26 k invalid location mentions. In the end, 2470 tweets contain at least one valid location reference matching the targeted time frame (30 August).

The polygon stacking technique that was presented by Schulz et al. [18] allows for combining multiple geographic information sources in view of supporting toponym resolution at the continental scale. In our work, we focus on a smaller area of interest, and resort to a simplified variant of this method in order to combine the spatial information provided by Twitter (Section 5.4) with location named entities that were extracted via NER. For a given tweet, the intersection of all bounding boxes is computed, with an index incremented, depending on the degree of overlap. In this context, the geotag adds an increment to all bounding boxes that it overlaps with. Figure 2 summarizes this procedure. For each tweet, we sort the resulting bounding boxes with respect to decreasing indexes. We retain the first resulting intersection with an associated surface less or equal to 20 km^2 (if it exists), in order to be consistent with the granularity requirements in this work. This procedure results in the selection of 5144 tweets, irrespective of the account of geotags. Hence, the amount of accounted tweets is doubled when jointly using NER and Twitter place bounding boxes, when compared to using Twitter place bounding boxes alone, as discussed in Section 5.4. We note that polygon stacking differs essentially from the latter section in the account of geotags: while they are converted to bounding boxes with dispersion $d_{\min}$ there, they merely increment the indexes of bounding boxes containing them here.

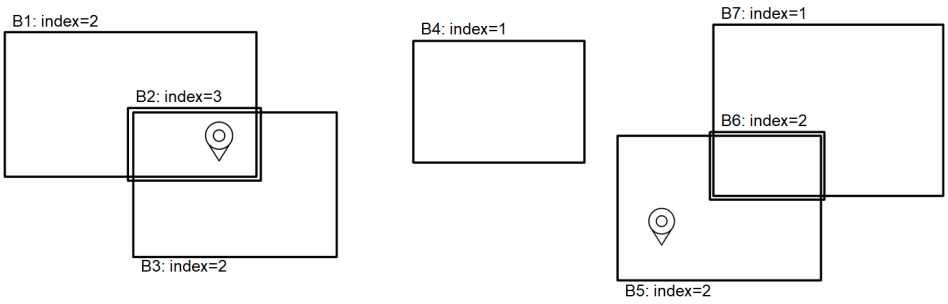

Figure 2. Simplified polygon stacking procedure. New bounding boxes are created at intersections, with increased index (for example B2 and B6). Geotags increment the index of all bounding boxes that include them.

6. Experiments

6.1. Experimental Protocol

As the functional form f in Equation (1), we use boosted ensembles of 10 logistic regression classifiers, optimized using the limited-memory BFGS algorithm [49]. This model outputs probabilities $y^*(p)$ that coordinates p belong to a flooded area. Let us define Δ_{map} the set of coordinates, so that $p \in \Delta_{\text{map}}$ if p is included in the area of the target map described in Section 5.1 . Training data couples $\{x(p), y(p)\}$ are then made by picking y values from the target map, which are associated to a feature vector built using a mapping function.

We define the support of a collection of tweets with respect to a weight function as the subset of Δ_{map}, so that the cumulated weight $\sum_n^N w_n(p, p_n, d_n)$ for coordinates p in this subset is greater than a threshold. As Figure 3 illustrates, this support generally lies in urban areas, from which tweets are generally sent. Coordinates outside this support area

do not have enough tweets in their vicinity to enable the reliable computation of a feature vector. In contrast to remote sensing based methods, prediction of flooded areas using social media will hence be limited to this support. We note that the support is a function of the tweet collection, which varies in our experiments depending on the geographic pre-processing method used (see Sections 5.4 and 5.5), reported in Table 1. As the table shows, it also depends on the weight function used to compute the feature vector mapping. We use $w_{\max}/2 = w(p,p,d_{\min})/2$ as threshold, meaning that p belongs to the support only if its cumulated weight is greater than half the weight obtained if p matches the center of a tweet bounding box with dispersion $d_{\min}$. In Table 1, we see that using geotags and polygon stacking yields an increased support. Masking pixels with the flood-fill algorithm naturally decreases this support (by approximately 50%). We note that limiting the support decreases the imbalance of the target variable y, and that flood-fill masking tends to reinforce this tendency.

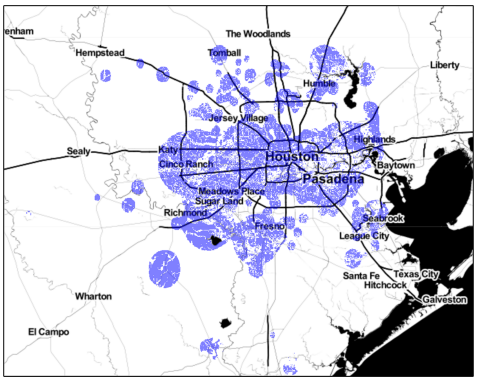

Figure 3. Support obtained for the polygon stacking spatial information extraction with w (in blue). The map tiles are by Stamen Design.

Table 1. Support and imbalance for the tested spatial information extraction methods.

Spatial Information	With w		With w^t	
	Support (# Pixels)	Imbalance	Support (# Pixels)	Imbalance
Place	49,077	11%	23,652	16%
Geotag	57,616	10%	29,405	11%
Place+geotag	83,514	10%	45,628	12%
Polygon stack	56,549	11%	28,444	15%

Each experiment that is reported in the next section is characterized by a weight function (w or w^t, see Section 4), a feature vector type (x or x^F, see Section 4), a textual representation (*keyword-matching*, *TFIDF*, or *Tweet2vec*, see Section 5.3) and a spatial information extraction method (place bounding boxes only, geotags only, place and geotag jointly, polygon stacking without geotags or polygon stacking with geotags). For each experiment, we extract a stratified random sample (with same balance as the original data set) of 200 k coordinates in Δ_{map} (23% of the map), compute the support of this subset with respect to the tested spatial information extraction method, and fit a function f^* using Equation (1) that was parametrized by the tested experimental conditions. We then use the whole support of Δ_{map} as the test set, so that the results are easily comparable across experimental conditions.

6.2. Quantitative and Qualitative Results

Table 2 reports the test precision, recall, and F1 (defined as the harmonic mean between precision and recall) scores for all possible experimental conditions. Metrics for *Top-10* predictions, which is the 10% most confident positive and negative predictions, are also reported. Each metric shown in Table 2 averages the results of five independent

experiments. Limiting to Top-10 predictions further reduces the predicted area, but outputs coordinates with higher confidence. Such information is still valuable in the context of assimilation approaches that can cope with partial maps [9].

Table 2. Test F1, precision, and recall scores for varying pre-processing and feature vector construction methods. The best results for each metric are emphasized.

Pre-Processing	Feature Vector	F1	P	R	Top-10 F1	Top-10 P	Top-10 R
Place + geotag	*keyword-match* + w^t + x^F	0.167	0.101	0.552	0.115	0.065	0.531
Place + geotag	*TFIDF*	0.319	0.273	0.386	0.684	0.561	0.878
	TFIDF + w^t	0.365	0.313	0.440	0.710	0.621	0.829
	TFIDF + x^F	0.272	0.184	0.517	0.485	0.339	0.859
	TFIDF + w^t + x^F	0.319	0.239	0.483	0.521	0.390	0.792
Place only	*TFIDF*	0.313	0.220	0.546	0.564	0.407	0.917
	TFIDF + w^t	0.404	0.305	0.603	0.777	0.665	0.937
	TFIDF + x^F	0.305	0.210	0.563	0.554	0.388	0.972
	TFIDF + w^t + x^F	0.413	0.320	0.586	0.799	0.674	**0.984**
Geotag only	*TFIDF*	0.327	0.320	0.336	0.682	0.633	0.739
	TFIDF + w^t	0.309	**0.388**	0.258	0.609	0.612	0.610
	TFIDF + x^F	0.287	0.216	0.431	0.586	0.454	0.830
	TFIDF + w^t + x^F	0.262	0.244	0.286	0.455	0.381	0.575
Poly. stack	*TFIDF*	0.333	0.272	0.432	0.658	0.549	0.821
	TFIDF + w^t	0.406	0.324	0.545	0.796	0.725	0.884
	TFIDF + x^F	0.315	0.232	0.496	0.623	0.482	0.882
	TFIDF + w^t + x^F	0.399	0.310	0.558	0.775	0.680	0.902
Poly. + geotag	*TFIDF*	0.325	0.252	0.459	0.658	0.527	0.879
	TFIDF + w^t	0.403	0.311	0.575	0.792	0.706	0.902
	TFIDF + x^F	0.307	0.221	0.502	0.597	0.448	0.898
	TFIDF + w^t + x^F	0.392	0.307	0.544	0.767	0.666	0.903
Place + geotag	*Tweet2Vec*	0.267	0.182	0.501	0.467	0.334	0.775
	Tweet2Vec + w^t	0.315	0.230	0.499	0.458	0.331	0.746
	Tweet2Vec + x^F	0.275	0.191	0.497	0.512	0.359	0.895
	Tweet2Vec + w^t + x^F	0.331	0.256	0.469	0.625	0.489	0.867
Place only	*Tweet2Vec*	0.301	0.207	0.547	0.533	0.379	0.900
	Tweet2Vec + w^t	0.410	0.314	0.600	0.748	0.634	0.912
	Tweet2Vec + x^F	0.310	0.222	0.518	0.578	0.413	0.965
	Tweet2Vec + w^t + x^F	**0.425**	0.342	0.567	**0.834**	**0.728**	0.977
Geotag only	*Tweet2Vec*	0.289	0.219	0.427	0.582	0.466	0.776
	Tweet2Vec + w^t	0.278	0.251	0.312	0.466	0.386	0.590
	Tweet2Vec + x^F	0.286	0.216	0.424	0.534	0.405	0.790
	Tweet2Vec + w^t + x^F	0.270	0.252	0.291	0.458	0.373	0.595
Poly. stack	*Tweet2Vec*	0.307	0.222	0.500	0.548	0.398	0.879
	Tweet2Vec + w^t	0.373	0.275	0.579	0.727	0.597	0.931
	Tweet2Vec + x^F	0.327	0.245	0.489	0.630	0.489	0.886
	Tweet2Vec + w^t + x^F	0.399	0.316	0.541	0.797	0.718	0.895
Poly. + geotag	*Tweet2Vec*	0.306	0.220	0.506	0.539	0.386	0.893
	Tweet2Vec + w^t	0.384	0.280	**0.610**	0.677	0.531	0.938
	Tweet2Vec + x^F	0.326	0.244	0.496	0.592	0.454	0.853
	Tweet2Vec + w^t + x^F	0.406	0.327	0.535	0.785	0.703	0.893

Only the best performing condition for the algorithm using the *keyword-matching* feature is reported. We see that its performance is very significantly below any other variant reported in Table 2. Furthermore, contrasting with all other tested conditions that are shown in Table 2, restricting to the 10% most confident predictions yields degraded performance metrics, which suggested that this setting is close to a fully random classifier. This observation highlights that merely matching a shortlist of carefully chosen keywords does not have enough expressiveness for the task.

First, we analyze the marginal influence of the flood-fill masked weights computation (w^t, see Equation (5)), the Fisher vector (x^F, see Equation (6)), and their combined use. With the geotag only spatial extraction method (for both textual representations), both w^t and x^F yield a degraded performance. In other cases, using w^t always yields very significant improvement in performance, on both global and Top-10 metrics. As noted earlier in Table 1, this improvement comes at the cost of a reduced support, though. Using the Fisher vector x^F occasionally brings improvement, but this tendency is less general than that observed while using w^t. With the *TFIDF* representation, except under the place only

spatial extraction method in conjunction with w^t, using x^F always leads to a performance degradation (especially in terms of precision). With the *Tweet2Vec* representation, the improvement with respect to the global metrics brought by x^F alone is not very significant, but it is much more apparent in terms of Top-10 metrics (for instance +0.082 Top-10 F1 with the polygon stacking method). Additionally, with the *Tweet2Vec* representation, a synergy between w^t and x^F is observed for most spatial extraction methods (for example, +0.086 Top-10 F1 with the place only condition).

Spatial extraction involving NER with polygon stacking methods does not yield spectacular improvement. However, they generally lead to decent performance (if gathering the Top 5 results for each metric column, 14 over 30 are tied to polygon stacking methods). In particular, they are generally associated with a good precision (for instance, the 2nd and 3rd best Top-10 precisions are obtained by polygon stacking variants). Actually, if the NER-based methods slightly underperform with respect to the place only methods (15 of the 30 best results), they are also tied to increased support (see Table 1): in comparison to using only Twitter place bounding boxes, we are able to infer approximately 20% more pixels at a minimal performance cost.

Polygon stacking yields stable results with or without using the geotags (seven over the best 30 results in each case). If we aggregate Table 2 with respect to the feature vector column, and then rank the polygon stacking variants, it appears that variants not using geotags are slightly ahead. If we focus on the F1 metric, which reflects precision and recall holistically, *Tweet2Vec* with w^t and x^F obtains both the most competitive overall F1 (0.425) and the best Top-10 F1 (0.834). Both of these results are established with the place only spatial extraction method.

On 31 August 2017, just one day after the S-1 acquisitions that were used to build the flood map in Figure 1a, a GeoEye-1 image was acquired over the Houston area and it was immediately made available thanks to Digital Globe's Open data program. The image clearly shows the massive flooding in and around Houston due to reduced cloud coverage at the acquisition time. In addition to GeoEye-1 images, by 31 August 2017, a total of 14,525 points over the city of Houston had been labeled as flooded houses or roads by Digital Globe's Tomnod crowdsourcing team (Data available for download at https://www.maxar.com/open-data/hurricane-harvey (accessed on 17 March 2021)), providing pointwise and independent interpretation of these images. Figure 4a,b show how predictions by our method (flooded in blue, non-flooded in grey) relate to the labeled points (marker glyphs). The displayed sub-figures were chosen not only to qualitatively show the overall agreement of our method with the ground truth labels, but also highlight the complementarity between these information sources, with expanded riverbeds inferred as flooded, but also grey areas marking predicted flooding limits.

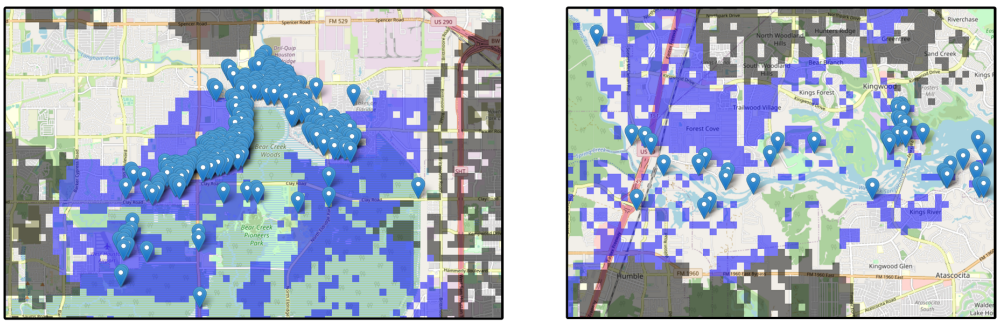

Figure 4. Qualitative comparison of raster cell predictions by our method (blue cells are flooded, grey non-flooded, obtained with polygon stacking, *Tweet2Vec*, x^t and x^F) and coordinates of buildings surrounded by floodwater extracted from a GeoEye-1 VHR imagery acquired on 31 August 2017 via crowdsourcing (marker glyphs). The areas shown are close to Addicks Reservoir (**a**) and Kingwood (**b**). The map tiles are by OpenStreetMap.

7. Discussion

The *TFIDF* representation for tweets is very sparse, and it generally gets better as more tweets are accumulated. Alternatively, the *Tweet2Vec* representation is meant to effectively represent tweets individually, but there is no guarantee that their weighted average is a fair representation of the whole collection. The differentiated effectiveness of the Fisher vector is explained by the fact it reflects the distribution of the weighted *Tweet2Vec* vectors with respect to the current coordinates, which is a more sensible way to exploit the individual representation capacity of the *Tweet2Vec* feature vector.

Regarding spatial extraction methods, it is interesting to note that most best-performing variants with respect to any metric reported in Table 2 do not use the geotag. If gathering the Top 5 results for each metric column (so, 30 results overall), only one makes primary use of the geotag (we did not include the polygon stacking + geotag method, as the geotag has a secondary utility in this case). Therefore, we provided experimental confirmation of the sensor displacement problem, coined by Robertson and Feick [23], which disqualifies Twitter geotags for fine-grained spatial applications, such as considered in this paper. We found out that exploiting the bounding boxes without consideration of the geotags led to the construction of more effective flood mapping functions, at the cost of reduced spatial support (see Table 1). This conclusion is further supported by the fact that the most competitive condition overall does not use geotags. Combining these bounding boxes to toponyms extracted thanks to NER techniques emerged as a middle path, with increased support at minimal influence on model performance.

Our Twitter corpus collection method is based on geographic criteria instead of keywords. This is meant to avoid false positive problems (for example, people posting support messages to population affected by Harvey from other US states) as well as low recall of the relevant content (many relevant posts may be missed with keyword searches). However, doing so, we retrieve many unrelated Twitter posts, for example, from automated sources. In the end, our experiments show that even a strict and small set of keywords yields low usability for our application. In contrast, using language models and identifying relevant features discriminatively acts as an automatic filter of unrelated content, even if this filter is far from perfect, as the experimental results show.

In this work, we circumvented the cost of individual tweet annotation by using aggregation with respect to a coordinate space, in an agnostic way regarding the topic of the content, discriminatively learning decisive features. Among types of content found on Twitter lies automatically generated content. If we consider this type of content as an additional set of topics, our method intrinsically copes with this content, just adding to the difficulty of the model fitting task. The only hypothesis that we rely on is that the spam proportion remains fairly constant through time. In parallel, the authors of the present study work on active learning aiming at efficient Twitter corpus labelling [57]. In preliminary experiments, the method was found to be effective for quickly isolating a large part of the automatically generated content from the remainder of the corpus. Filtering the set of selected tweets using this method could be tested in order to further improve the mapping quality.

The present approach assumes that the representativeness of the general population by the set of Twitter users is constant with respect to geographic coordinates. A potential bias regarding wealth in the Twitter population may severely harm the validity of this hypothesis. A slight over-representation of wealthy classes in the Twitter user base seems to exist, but the penetration rate is similar in the underprivileged and middle classes [58].

Experimentally, we found that areas with sufficient support are heavily correlated to the population density. Assuming that the proportion of Twitter users is constant with respect to space, this is a quite natural conclusion. As a consequence, the practical usability of the present method is restricted to urban areas. The supported area may be extended by lowering the threshold presented in Section 6.1, but experimentally doing so led to systematic performance degradation.

The take-away message to stakeholders, such as emergency response or civil protection authorities, is that regional event mapping using Twitter from the implicit VGI perspective is possible in an automated way, but they should be aware of the restriction to sufficiently populated areas, as only a small fraction of the Twitter feed is usable in practice. An alternative (or complementary) way could be the move to explicit VGI, for example, with incentives for population to send localized storm-related content that is associated to a dedicated hashtag. However, anticipation is needed in the latter case, which is often impossible, especially in the case of unpredictable events, such as riots.

8. Conclusions

The proposed classification model converts a collection of tweets to flood probabilities for the subset of a map, determined by the presence of a sufficient support in terms of Twitter content. We emphasize that this support correlates to urban density. Extensive experiments show that combining filtering with respect to a DEM and Fisher vectors yields increased performance, and neural embedding models bring a significant qualitative improvement. At best, our method is able to reach 0.425 F1 score if predicting the whole map support, and 0.834 when restricting to the 10% most confident predictions.

Restricting the predicted area to the most confident model outputs yields an important boost. We quantified to which extent the various experimental conditions affect the size of the predicted area as well as the classification accuracy. We note that some applications, such as data assimilation, can cope with rather sparse areas, provided that the accuracy of predictions is high enough. Our most immediate perspective is to validate the proposed method in the context of a data assimilation technique for flood forecasting, such as presented in Sections 1 and 2, hence assessing the value brought by an observation proxy with higher temporal frequency.

Tweets also often feature images, some of which may picture flooded scenes. Detecting flooded scenes has already been considered in the literature. For example, the MediaEval Multimedia Satellite task disclosed a dedicated training data set [59]. As a perspective, we could consider including this information as additional input to the feature mapping function that is described in Section 4.

The procedure that is described in this paper requires training the model using the data of a reference day. The model is then used in production for any time frame of another day to generate novel maps. We believe that our approach is robust to changes in distribution of the messages sent on Twitter, but probably not to the appearance of novel topics, for example, if a specific public building or bridge collapses due to an ongoing storm. Beyond simply updating the model on a regular basis, the evolution of our model to a stochastic process integrating time could address this limitation.

Author Contributions: Conceptualization, P.B., E.B. and S.M.-M.; Methodology, Formal Analysis, P.B.; Software, Data Curation, P.B. and E.B.; Writing—Original Draft Preparation, P.B.; Writing—Review & Editing, E.B., S.M.-M., R.H., M.C., R.-M.P.; Supervision, P.M.; Project Administration, Funding Acquisition, T.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Luxembourgish National Research Fund (FNR) Publimape project, grant number C16/IS/11335103.

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

References

1. Sakaki, T.; Okazaki, M.; Matsuo, Y. Tweet Analysis for Real-Time Event Detection and Earthquake Reporting System Development. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 919–931.
2. de Bruijn, J.; de Moel, H.; Jongman, B.; Wagemaker, J.; Aerts, J. TAGGS: Grouping Tweets to Improve Global Geoparsing for Disaster Response. *J. Geovisualiz. Spat. Anal.* **2017**, *2*, doi:10.1007/s41651-017-0010-6.
3. Chen, L.; Butler, P.; Ramakrishnan, N.; Prakash, B. Syndromic surveillance of Flu on Twitter using weakly supervised temporal topic models. *Data Min. Knowl. Discov.* **2016**, *30*, 681–710.
4. Jongman, B.; Wagemaker, J.; Romero, B.; De Perez, E. Early Flood Detection for Rapid Humanitarian Response: Harnessing Near Real-Time Satellite and Twitter Signals. *ISPRS Int. J. Geo-Inf.* **2015**, *4*, 2246–2266.

5. De Groeve, T.; Riva, P. Global real-time detection of major floods using passive microwave remote sensing. In Proceedings of the 33rd International Symposium on Remote Sensing of Environment, Tucson, AZ, USA, 4–8 May 2009.
6. Wiegmann, M.; Kersten, J.; Senaratne, H.; Potthast, M.; Klan, F.; Stein, B. Opportunities and Risks of Disaster Data from Social Media: A Systematic Review of Incident Information. In *Natural Hazards and Earth System Sciences Discussions*; [preprint under review]; Copernicus Publications: Göttingen, Germany, 2020; pp. 1–16.
7. Revilla-Romero, B.; Wanders, N.; Burek, P.; Salamon, P.; de Roo, A. Integrating remotely sensed surface water extent into continental scale hydrology. *J. Hydrol.* **2016**, *543*, 659–670.
8. Grimaldi, S.; Li, Y.; Pauwels, V.R.N.; Walker, J.P. Remote Sensing-Derived Water Extent and Level to Constrain Hydraulic Flood Forecasting Models: Opportunities and Challenges. *Surv. Geophys.* **2016**, *37*, 977–1034.
9. Hostache, R.; Chini, M.; Giustarini, L.; Neal, J.; Kavetski, D.; Wood, M.; Corato, G.; Pelich, R.M.; Matgen, P. Near-Real-Time Assimilation of SAR-Derived Flood Maps for Improving Flood Forecasts. *Water Resour. Res.* **2018**, *54*, 5516–5535.
10. MacEachren, A.M.; Jaiswal, A.; Robinson, A.C.; Pezanowski, S.; Savelyev, A.; Mitra, P.; Zhang, X.; Blanford, J. SensePlace2: GeoTwitter analytics support for situational awareness. In Proceedings of the 2011 IEEE Conference on Visual Analytics Science and Technology (VAST), Providence, RI, USA, 23–28 October 2011; pp. 181–190.
11. Crooks, A.; Croitoru, A.; Stefanidis, A.; Radzikowski, J. #Earthquake: Twitter as a Distributed Sensor System. *Trans. GIS* **2013**, *17*, 124–147.
12. Cheng, T.; Wicks, T. Event Detection using Twitter: A Spatio-Temporal Approach. *PLoS ONE* **2014**, *9*, e97807.
13. Craglia, M.; Ostermann, F.; Spinsanti, L. Digital Earth from vision to practice: Making sense of citizen-generated content. *Int. J. Digit. Earth* **2012**, *5*, 398–416.
14. Middleton, S.; Middleton, L.; Modafferi, S. Real-Time Crisis Mapping of Natural Disasters Using Social Media. *IEEE Intell. Syst.* **2014**, *29*, 9–17.
15. Granell, C.; Ostermann, F.O. Beyond data collection: Objectives and methods of research using VGI and geo-social media for disaster management. *Comput. Environ. Urban Syst.* **2016**, *59*, 231–243.
16. Zhang, C.; Fan, C.; Yao, W.; Hu, X.; Mostafavi, A. Social media for intelligent public information and warning in disasters: An interdisciplinary review. *Int. J. Inf. Manag.* **2019**, *49*, 190–207.
17. Grace, R. Hyperlocal Toponym Usage in Storm-related Social Media. In Proceedings of the 17th ISCRAM Conference, Blacksburg, VA, USA, 24–27 May 2020.
18. Schulz, A.; Hadjakos, A.; Paulheim, H.; Nachtwey, J.; Mühlhäuser, M. A Multi-Indicator Approach for Geolocalization of Tweets. In Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media, Dublin, Ireland, 4–7 June 2013; pp. 1–10.
19. Dittrich, A. Real-Time Event Analysis and Spatial Information Extraction From Text Using Social Media Data. Ph.D. Thesis, KIT, Karlsruher, Germany, 2016.
20. Mishra, S.; Diesner, J. Semi-supervised Named Entity Recognition in noisy-text. In Proceedings of the 2nd Workshop on Noisy User-Generated Text (WNUT), Osaka, Japan, 11 December 2016; pp. 203–212.
21. Krieger, M.; Ahn, D. TweetMotif: Exploratory search and topic summarization for Twitter. In Proceedings of the AAAI Conference on Weblogs and Social Media, Washington, DC, USA, 23–26 May 2010.
22. Brangbour, E.; Bruneau, P.; Marchand-Maillet, S.; Hostache, R.; Matgen, P.; Chini, M.; Tamisier, T. Extracting localized information from a Twitter corpus for flood prevention. *arXiv* **2019**, arXiv:1903.04748.
23. Robertson, C.; Feick, R. Inference and analysis across spatial supports in the big data era: Uncertain point observations and geographic contexts. *Trans. GIS* **2018**, *22*, 455–476.
24. Bates, P.; De Roo, A. A simple raster-based model for flood inundation simulation. *J. Hydrol.* **2000**, *236*, 54–77.
25. Andreadis, K.M.; Schumann, G.J.P. Estimating the impact of satellite observations on the predictability of large-scale hydraulic models. *Adv. Water Resour.* **2014**, *73*, 44–54.
26. García-Pintado, J.; Mason, D.C.; Dance, S.L.; Cloke, H.L.; Neal, J.C.; Freer, J.; Bates, P.D. Satellite-supported flood forecasting in river networks: A real case study. *J. Hydrol.* **2015**, *523*, 706–724.
27. Brouwer, T.; Eilander, D.; Van Loenen, A.; Booij, M.; Wijnberg, K.; Verkade, J.; Wagemaker, J. Probabilistic flood extent estimates from social media flood observations. In *Natural Hazards and Earth System Sciences*; Copernicus Publications: Göttingen, Germany, 2017; Volume 17.
28. Nobre, A.D.; Cuartas, L.A.; Hodnett, M.; Rennó, C.D.; Rodrigues, G.; Silveira, A.; Waterloo, M.; Saleska, S. Height Above the Nearest Drainage—A hydrologically relevant new terrain model. *J. Hydrol.* **2011**, *404*, 13–29.
29. Nobre, A.D.; Cuartas, L.A.; Momo, M.R.; Severo, D.L.; Pinheiro, A.; Nobre, C.A. HAND contour: A new proxy predictor of inundation extent. *Hydrol. Process.* **2016**, *30*, 320–333.
30. Eilander, D.; Trambauer, P.; Wagemaker, J.; van Loenen, A. Harvesting Social Media for Generation of Near Real-time Flood Maps. *Procedia Eng.* **2016**, *154*, 176–183.
31. Karssenberg, D.; Burrough, P.; Sluiter, R.; de Jong, K. The PCRaster Software and Course Materials for Teaching Numerical Modelling in the Environmental Sciences. *Trans. GIS* **2001**, *5*, 99–110.
32. Fohringer, J.; Dransch, D.; Kreibich, H.; Schröter, K. Social media as an information source for rapid flood inundation mapping. *Nat. Hazards Earth Syst. Sci.* **2015**, *15*, 2725–2738.

33. Joachims, T. Text categorization with Support Vector Machines: Learning with many relevant features. In *ECML-98*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 137–142.
34. Lampos, V.; Cristianini, N. Nowcasting Events from the Social Web with Statistical Learning. *ACM Trans. Intell. Syst. Technol.* **2012**, *3*, 1–22.
35. Dhingra, B.; Zhou, Z.; Fitzpatrick, D.; Muehl, M.; Cohen, W. Tweet2Vec: Character-Based Distributed Representations for Social Media. *arXiv* **2016**, arXiv:1605.03481.
36. Oh Song, H.; Xiang, Y.; Jegelka, S.; Savarese, S. Deep Metric Learning via Lifted Structured Feature Embedding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4004–4012.
37. Xiang, G.; Fan, B.; Wang, L.; Hong, J.; Rose, C. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management. Association for Computing Machinery, Maui, HI, USA, 29 October 2012; pp. 1980–1984.
38. Parekh, P.; Patel, H. Toxic Comment Tools: A Case Study. *Int. J. Adv. Res. Comput. Sci.* **2017**, *8*, 964–967.
39. Gao, Y.; Wang, S.; Padmanabhan, A.; Yin, J.; Cao, G. Mapping spatiotemporal patterns of events using social media: A case study of influenza trends. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 425–449.
40. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
41. Graves, A.; Fernández, S.; Schmidhuber, J. Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In *Artificial Neural Networks: Formal Models and Their Applications—ICANN 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 799–804.
42. Littman, J. Hurricanes Harvey and Irma Tweet ids. 2017. Available online: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/QRKIBW (accessed on 17 March 2021)
43. Brangbour, E.; Bruneau, P.; Marchand-Maillet, S.; Hostache, R.; Chini, M.; Matgen, P.; Tamisier, T. Computing flood probabilities using Twitter: Application to the Houston urban area during Harvey. In Proceedings of the 9th International Workshop on Climate Informatics, Paris, France, 2–4 October 2019.
44. Krapac, J.; Verbeek, J.; Jurie, F. Modeling spatial layout with fisher vectors for image categorization. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1487–1494.
45. Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Singapore, 2–7 August 2009; pp. 1003–1011.
46. Giustarini, L.; Hostache, R.; Kavetski, D.; Chini, M.; Corato, G.; Schlaffer, S.; Matgen, P. Probabilistic Flood Mapping Using Synthetic Aperture Radar Data. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6958–6969.
47. Chini, M.; Pelich, R.; Pulvirenti, L.; Pierdicca, N.; Hostache, R.; Matgen, P. Sentinel-1 InSAR Coherence to Detect Floodwater in Urban Areas: Houston and Hurricane Harvey as A Test Case. *Remote Sens.* **2019**, *11*, 107.
48. Pulvirenti, L.; Chini, M.; Pierdicca, N. InSAR Multitemporal Data over Persistent Scatterers to Detect Floodwater in Urban Areas: A Case Study in Beletweyne, Somalia. *Remote Sens.* **2021**, *13*, 37.
49. Fletcher, R. *Practical Methods of Optimization*, 2nd ed.; Wiley & Sons: Hoboken, NJ, USA, 1987.
50. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2017**, arXiv:1609.04747.
51. Lampos, V.; Zou, B.; Cox, I. Enhancing Feature Selection Using Word Embeddings: The Case of Flu Surveillance. In Proceedings of the 26th International Conference on World Wide Web, Perth Australia, 8 April 2017; pp. 695–704.
52. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *J. R. Stat. Soc. Ser. B Methodol.* **1977**, *39*, 1–22.
53. Porter, M. An algorithm for suffix stripping. *Program* **1980**, *14*, 130–137.
54. Kitamoto, A.; Sagara, T. Toponym-based geotagging for observing precipitation from social and scientific data streams. In Proceedings of the ACM Multimedia 2012 Workshop on Geotagging and Its Applications in Multimedia, Nara, Japan, 2 November 2012; pp. 23–26.
55. Fung, I.C.H.; Tse, Z.T.H.; Cheung, C.N.; Miu, A.S.; Fu, K.W. Ebola and the social media. *Lancet* **2014**, *384*, 2207–2207.
56. Shelton, T.; Poorthuis, A.; Graham, M.; Zook, M. Mapping the data shadows of Hurricane Sandy: Uncovering the sociospatial dimensions of 'big data'. *Geoforum* **2014**, *52*, 167–179.
57. Brangbour, E.; Bruneau, P.; Tamisier, T.; Marchand-Maillet, S. Active Learning with Crowdsourcing for the Cold Start of Imbalanced Classifiers. In Proceedings of the 17th International Conference on Cooperative Design, Visualization, and Engineering, Whistler, BC, Canada, 25–28 October 2020; pp. 192–201.
58. Perrin, A.; Anderson, M. Share of U.S. Adults Using Social Media, Including Facebook, Is Mostly Unchanged since 2018. Pew Research Center. 2019. Available online: https://www.pewresearch.org/fact-tank/2019/04/10/share-of-u-s-adults-using-social-media-including-facebook-is-mostly-unchanged-since-2018/ (accessed on 17 March 2021.)
59. Bischke, B.; Helber, P.; Schulze, C.; Srinivasan, V.; Dengel, A.; Borth, D. The Multimedia Satellite Task at MediaEval 2017. In Proceedings of the MediaEval 2017 Workshop, Dublin, Ireland, 13–15 September 2017.