ARTICLE

Dynamic Spatio-Temporal Tweet Mining for Event Detection: A Case Study of Hurricane Florence

Mahdi Farnaghi¹ · Zeinab Ghaemi² · Ali Mansourian^{1,3}

Published online: 26 May 2020
© The Author(s) 2020

Abstract Extracting information about emerging events in large study areas through spatiotemporal and textual analysis of geotagged tweets provides the possibility of monitoring the current state of a disaster. This study proposes dynamic spatio-temporal tweet mining as a method for dynamic event extraction from geotagged tweets in large study areas. It introduces the use of a modified version of ordering points to identify the clustering structure to address the intrinsic heterogeneity of Twitter data. To precisely calculate the textual similarity, three state-of-the-art text embedding methods of Word2vec, GloVe, and FastText were used to capture both syntactic and semantic similarities. The impact of selected embedding algorithms on the quality of the outputs was studied. Different combinations of spatial and temporal distances with the textual similarity measure were investigated to improve the event detection outcomes. The proposed method was applied to a case study related to 2018 Hurricane Florence. The method was able to precisely identify events of varied sizes and densities before, during, and after the hurricane. The feasibility of the proposed method was qualitatively evaluated using the Silhouette coefficient and qualitatively discussed. The proposed method was also compared to an implementation based on the standard density-based spatial

clustering of applications with noise algorithm, where it showed more promising results.

Keywords Disaster management · Hurricane Florence · Natural language processing · Spatio-temporal tweet analysis · Tweet clustering · Twitter

1 Introduction

Development and proliferation of social networks, as well as their popularity, provides the possibility for users to play a new role as social sensors who observe different events and publish their understanding and opinion about the social and natural events that they witness. These human sensors dynamically share messages related to a wide variety of topics, while the use of mobile devices equipped with positioning sensors enriches the messages with spatiotemporal information. The opportunities provided by these geotagged messages make social networks a potential source of information in different domains, and particularly disaster management. Analysis of the spatiotemporal distribution of messages, while considering their textual content, using cluster detection methods can extract groups of geotagged messages that highlight particular issues before, during, and after a disaster. Such information is extremely valuable at different stages of emergencies (Krajewski et al. 2016; Sit et al. 2019). Remarkably in the response phase, the location of emergency situations, as well as the damages, can be captured (Farnaghi and Mansourian 2013; Sit et al. 2019) in real time from clusters of similar tweets that are published by individuals who have witnessed the same event. The extracted information can help planners and disaster managers to implement

✉ Mahdi Farnaghi
mahdi.farnaghi@nateko.lu.se

¹ GIS Center, Department of Physical Geography and Ecosystem Science, Lund University, 223 62 Lund, Sweden

² Faculty of Geodesy and Geomatics, K.N. Toosi University of Technology, Tehran, Iran

³ Center for Middle Eastern Studies, Lund University, 221 00 Lund, Sweden

appropriate measures and intervention plans to deal with the incidents and alleviate their consequences.

Twitter, as the most popular microblogging social network, has been widely used for online event detection (Hasan et al. 2018). In this context, the three components of location, time, and content of tweets should be considered for event detection using Twitter data. While the contents of the geotagged tweets aid in determining the nature of the events, their spatial positions help with detecting the locations of the events and risky areas in disaster management, and their time stamps assist in identifying the duration of the events. Previous studies have had different approaches to deal with these three dimensions. A large body of studies has merely considered the textual content of tweets for event detection and disregarded the two other aspects of time and location (Huang and Xiao 2015; Kirilenko and Stepchenkova 2017; Sriji et al. 2017; Sutton et al. 2018; Niederkrotenthaler et al. 2019). There have also been studies that used spatial analysis methods in addition to keyword-based filtering or textual analysis to extract the location of events (Steiger et al. 2015; Yang and Mu 2015; Cui et al. 2017; Nguyen and Shin 2017; Nguyen et al. 2017; Ghaemi 2019).

In the last decade, researchers have started considering both spatial and temporal dimensions to reveal the hidden patterns of Twitter data. Some of these efforts either neglected the textual content of the tweets or simply filtered the input tweets using keywords related to the interesting events (Cheng and Wicks 2014; Wang et al. 2016). Other efforts have focused on taking the spatial, temporal, and textual aspects of Twitter data into account, either by analyzing the spatiotemporal dimensions and textual dimension in two separate steps—see, for example, the proposed real-time event detection system by Walther and Kaisser (2013) and Geo-H-SOM by Steiger et al. (2016)—or by applying clustering algorithms to simultaneously analyze the three components of location, time, and textual content (Croitoru et al. 2015; Capdevila et al. 2017).

Clustering algorithms are powerful unsupervised approaches that divide the entire dataset into groups of similar objects. Clustering of tweets—considering their content, location, and time during a disaster—results in groups of tweets with similar content that are close together in space and time, and mostly refer to the events that are witnessed in the same area. Various clustering algorithms, including hierarchical (Kaleel and Abhari 2015), partitioning (Vijayarani and Jothi 2014), and density-based (Liu et al. 2007; Ben-Lhachemi and Nfaoui 2018) have been utilized for event detection from geotagged tweets. Among them, density-based algorithms—especially the density-based spatial clustering of applications with noise (DBSCAN) and its variations—are the most commonly used approaches (Arcaini et al. 2016; Capdevila et al.

2017) due to their ability in detecting clusters with arbitrary shapes, while not being sensitive to noisy datasets. Moreover, DBSCAN does not require prior knowledge of the number of clusters (Ester et al. 1996; Parimala et al. 2011).

In this context, Arcaini et al. (2016) used an approach based on filtering and an extended DBSCAN algorithm, named GT-DBSCAN, to reveal the geo-temporal structure of interesting events from Twitter data. Croitoru et al. (2015) used DenStream, a density-based clustering algorithm for streaming data, to extract spatiotemporal events from Twitter data while considering user groups and their relationships. GDBSCAN, another extension of DBSCAN (Sander et al. 1998), was exploited by Capdevila et al. (2017) for event extraction from Twitter data, based on the content, time, location, and publishers of tweets. Two other descendants of DBSCAN, named ST-DBSCAN and IncrementalDBSCAN, were also utilized for spatiotemporal clustering and event detection from Twitter data by Huang et al. (2018) and Lee (2012), respectively.

1.1 Problem Statement

Regarding spatial, temporal, and textual aspects of tweets, previous studies have been able to successfully address several problems in the context of event detection from geotagged tweets. However, despite their advantages, most of the density-based clustering algorithms like DBSCAN and its branches do not account for the spatial heterogeneity of the Twitter data. They use global input parameters for the whole study area, which prevents the algorithms from extracting local clusters with varied densities (Idrissi et al. 2015). This problem is magnified when the method is going to extract local events in large geographical areas where there are various demographic locations with different population densities that are affected by different events of varying importance. Local events that often occur during or after a disaster—for example, power outages or fires—lead to an increase in the damages and even casualties. Detection of these local events across a large study area requires adjusting input parameters based on the density of geotagged tweets for each area. But these parameters are hard to determine, especially when the input dataset is unknown or dynamically changing, as is the case for Twitter data. Moreover, the proposed solutions for determining these parameters, for example by Schubert et al. (2017), are not algorithmic and require human intervention.

Another issue of the previous solutions is related to the way they have modeled the distance between tweets by considering the locational, temporal, and textual dimensions of tweets in the clustering algorithm. While calculating the spatial and temporal distances among tweets is straightforward, calculating the textual similarities between

tweets is complicated and requires the utilization of natural language processing (NLP) techniques. In order to model the textual similarity among tweets, previous studies have overused traditional, frequency-based vectorization methods like count vector (CV) (Lee et al. 2011; Fócil-Arias et al. 2017), term frequency (TF) (Hecht et al. 2011), and term frequency inverse document frequency (TFIDF) (Phelan et al. 2009; Benhardus and Kalita 2013) to convert the textual contents of tweets into numerical vectors and then calculate the distance between those vectors. The problem of these frequency-based methods is that they result in huge vectors for representing the tweets. They also neglect the effect of synonyms/antonyms, the context, and the semantics of the texts. They are not capable of modeling the abbreviations and misspelled words that are frequently used in tweets. Moreover, considering the short length of tweets (no more than 280 characters), the output vectors of these methods are unfavorably sparse, which in turn hinders the feasibility of using distance functions like cosine distance for calculating the similarity among tweets. Another important issue in this regard is the requirement for evaluating the methods by which the three aspects of spatial, temporal, and textual content are combined to define an overall metric to present the distance between geotagged tweets. Proper definition of such metric directly affects the accuracy of the clustering algorithm.

These issues prevent us from having a system for disaster management that can dynamically detect spatiotemporal emergency events with varying densities in large-scale areas without human intervention. With current methods, we need specialists to tune the parameters of the event detection models and run them locally, based on the size and extent of the prospective events. The existing methods also have problems in detecting tweets that are similar in meaning and semantics but different in wording and syntactic structure. Hence, developing an efficient method that can overcome the mentioned problems can accelerate real-time event detection and facilitate disaster management.

1.2 Research Objectives

The main objective of this study is to propose a method, called dynamic spatio-temporal tweet mining (DSTTM), for event extraction from dynamic, real-time, geotagged Twitter data in large study areas of spatial heterogeneity without human intervention for disaster management. DSTTM receives geotagged tweets of the specified study area and uses unsupervised machine learning (ML) clustering algorithms and NLP to identify events as spatiotemporal clusters, visualize those clusters, and present them for further analysis by the disaster managers. DSTTM can be employed as a means to receive near real-time

knowledge about the nature of the disaster and its relative local events, as well as the way people look at and perceive those incidents. The proposed method has three defining characteristics:

1. The ability to address the spatial heterogeneity in Twitter data and sensitivity to the changes in the density of tweets in different locations;
2. The ability to consider spatial and temporal distances along with textual similarity in real-time extraction of spatiotemporal clusters;
3. The utilization of advanced NLP techniques, especially vectorization and text embedding methods, for calculating the textual similarities of tweets while considering the semantic similarities among tweets.

The following section is dedicated to materials and methods. The results are presented in Sect. 3, followed by a discussion in Sect. 4, and some future directions are proposed in the conclusion.

2 Materials and Methods

Hurricane Florence, an Atlantic hurricane in September 2018 that caused disastrous damage on the southeast seaboard of the United States, was selected as the case study. The geotagged tweets during the occurrence of the hurricane, from 12 September to 19 September 2018, were collected and used for a geographical area covering the two U.S. states of North Carolina and South Carolina (minimum longitude: -84.4341 , minimum latitude: 33.6761 , maximum longitude: -75.2556 , and maximum latitude: 36.6131). The events related to this hurricane, extracted from geotagged tweets, are mainly reported and discussed in this study.

2.1 The Dynamic Spatio-Temporal Tweet Mining Method

To be able to dynamically and autonomously extract events from Twitter data in a large study area with no prior knowledge of the content, location, and times of the tweets, DSTTM requires the use of a clustering algorithm that works with a minimum number of input parameters. To overcome the problem of heterogeneity in Twitter data that are continuously collected for a large geographical area, the algorithm should be sensitive to the changes in the density of the tweets in different locations. To fulfil these requirements, the ordering points to identify the clustering structure (OPTICS) approach was selected, modified, and used as the underlying clustering algorithm of DSTTM. OPTICS, an extension of DBSCAN, solves the shortcoming of DBSCAN in defining input parameters and

extracting clusters with varied densities in heterogeneous environments (Reddy and Ussenaiah 2012; Joshi and Kaur 2013).

In order to properly model the distance between geotagged tweets, we tested different formulas for combining spatial distance, temporal distance, and textual similarity into a single metric that can measure the ultimate distance between tweets. The best metric was used as the underlying metric in DSTTM.

Considering the shortcomings of simple vectorization methods, such as TF, TFIDF, and CV, for vectorization of short Twitter messages, three state-of-the-art text embedding algorithms of Word2Vec (Mikolov et al. 2013), GloVe (Pennington et al. 2014), and FastText (Bojanowski et al. 2017) were used in DSTTM. These algorithms, proposed by Google, the Stanford NLP Group, and Facebook, work based on a Deep Neural Network and provide the possibility to accurately calculate the textual similarities among tweets while considering the semantics of the texts.

Figure 1 shows the overall workflow of a prototype system that was developed based on DSTTM to be able to describe and test the method. The system runs in two independent execution processes.

The main goal of the first execution process is to alter the texts of tweets to analyzable texts. Whenever a new tweet is received by the Twitter Streaming application

programming interface (API), its text is transferred to lowercase, while URLs, special characters, and numbers are removed, the punctuation signs are deleted, and hash-tags are replaced by their text. Then the text is tokenized, the words are corrected for repeating characters, the stop words are removed, and the words are lemmatized. Finally, the lemmatized words are joined together and represented as a cleaned tweet. The processed text is saved in a spatial database as a point with its locational, temporal, and textual information.

The second execution process focuses on near real-time analysis of geotagged tweets that have been preprocessed and saved in the database. However, real-time and near real-time analysis of geotagged tweets for a large geographical area requires an appropriate strategy to deal with a huge amount of accumulative data. It is impossible to analyze the whole tweets that are progressively stored in the database, due to the restrictions of the memory and the processing power of the underlying hardware infrastructures. To address this issue, DSTTM adopts a sliding windows approach proposed by Bifet (2010) and used by Lee (2012). Figure 1 mentions the iterative nature of the second process in which the analyses are run in consecutive sliding windows. Starting from an initial time, $t = t_0$, in each iteration, the data related to the specified time window, between $[t - l, t]$, is retrieved from the database and

Fig. 1 The overall architecture of dynamic spatio-temporal tweet mining (DSTTM)

processed by the event detection procedure. The results are sent to the post-processing analyses, and finally, the outputs are visualized and evaluated. In the next iteration, the time window moves by δt , and the process is repeated for the new time window.

In each iteration, event detection starts by applying the OPTICS clustering algorithm on the data of the time window. In order to be able to analyze the effect of different vectorization and text embedding methods, five different methods—CV, TFIDF, Word2vec, GloVe, and FastText—were implemented in the system. Moreover, two different metrics were defined and used in the system to combine the spatial distance, temporal distance, and text similarity of tweets using weighted sum and multiplication operations (Sect. 2.2).

In every iteration, the cluster detection mechanism detects the clusters based on the content and spatiotemporal distances of tweets. These clusters represent the events that are observed at different locations in the study area and in the current time window. However, a spatiotemporal event detection system needs to be able to monitor and track a particular event over both time and space. While event detection in every iteration provides the ability to distinguish different events within a time window, the next step of the event detection process is to link the detected clusters at each location and time window to the clusters that were detected at that location in the previous iteration (time window). This requirement is addressed by linking clusters in consecutive iterations based on the temporal overlaps between the sliding time windows (Sect. 2.3).

Having the events detected as clusters by the event detection modules, the next step is to post-process the results. In this step, we need to extract a topic for each cluster (Sect. 2.4) and then calculate the Silhouette coefficient, which shows the quality of the clustering process (Sect. 2.5).

Finally, the events that are detected at each iteration are presented in 2-dimensional maps and 3-dimensional charts where the third axis represents time. Additionally, the word cloud of each cluster is generated based on the TFIDF method to better represent the textual content of the tweets in each cluster, and the shapes of the clusters are extracted by fitting confidence ellipsoids to the points of each cluster in 2-dimensional space.

2.2 Cluster Detection

Having P as a collection of geotagged tweets in the database, each tweet $p \in P$ is represented as a tuple $[x, y, t, c, l]$, where x and y are the geographical coordinates, t is the time stamp, c is the textual content, and l is the cluster label, which is undefined at the beginning.

2.2.1 Clustering Algorithm

DSTTM utilizes the OPTICS density-based clustering algorithm, which can deal with the heterogeneity in the data by detecting clusters of various sizes and density. In contrast to DBSCAN, which uses a binary indicator of density, OPTICS exploits a continuous indicator. It first generates an order list of input objects (called cluster order) so that the closest objects are neighbors on the list. Different algorithms, like the one by Schubert and Gertz (2018), can be used afterward to detect clusters from the ordered list.

OPTICS receives two parameters of $\min Pts$ and ϵ , where ϵ is the maximum radius to be considered for clustering, and $\min Pts$ is the minimum number of objects that must exist around an object so that those objects together can be considered as a cluster. In a loop, the algorithm randomly selects an unprocessed object as the current object and calculates the core distance of that object using Eq. 1.

$$\begin{aligned} CoreDist_{\epsilon, \min Pts}(p) &= \begin{cases} Undefined, & |\{p', Dist(p, p') \leq \epsilon\}| < \min Pts \\ \min Pts Dist(p), & otherwise \end{cases} \end{aligned} \quad (1)$$

If the core distance is not undefined, the successive neighborhoods of the object are traversed, and the reachability distance between the object and each of the neighbors is computed using Eq. 2.

$$\begin{aligned} ReachDist_{\epsilon, \min Pts}(p, o) &= \begin{cases} Undefined, & |\{p', Dist(p, p') \leq \epsilon\}| < \min Pts \\ \max(CoreDist(p), Dist(p, o)), & otherwise \end{cases} \end{aligned} \quad (2)$$

At this stage in the loop, the current object is added to the cluster order list; the neighbors of the current object are sorted based on their minimum reachability distance and added to the cluster order list; both the current object and its neighbors are considered as processed objects. When all the objects are processed within the loop, we have an ordered list in which the denser objects are listed beside each other. Plotting the ordered list on a graph where the x-axis shows the order and the y-axis depicts the reachability distance will show the clusters as valleys with deeper valleys pointing to denser clusters.

In this study, the OPTICS algorithm was implemented so that it can calculate the cluster order. Having the cluster order, the algorithm presented by Schubert and Gertz (2018) was used to extract clusters from the cluster order. Hence, the event detection procedure receives the collection of tweets for the current time window P_t as input and return P'_t , so that every tweet in the result set, $p' \in P'_t$, has a

defined cluster label, $p'.l = \text{cluster label}$, or its cluster label is set to *noise*, $p'.l' = \text{noise}$.

An important issue of the utilization of the OPTICS algorithm in this study was to define the distance metric, $\text{Dist}(p, q)$, so that it can consider the spatial and temporal proximity among tweets as well as their textual similarity.

2.2.2 Distance Metric

Two different distance metrics based on the weighted sum and multiplication of the spatial and temporal distances, and the textual similarity measure were defined, as presented in Eqs. 3 and 4, respectively:

$$\begin{aligned} \text{Dist}(p, q) = & \alpha \times \text{NormEuclDistSpatial}(p, q) + \beta \\ & \times \text{NormEuclDistTemporal}(p, q) + \gamma \\ & \times \text{TextualSim}(p, q) \end{aligned} \quad (3)$$

$$\begin{aligned} \text{Dist}(p, q) = & \text{EuclDistSpatialTemporal}(p, q) \\ & \times \text{TextualSim}(p, q) \end{aligned} \quad (4)$$

While in Eq. 3, $\text{NormEuclDistSpatial}$ and $\text{NormEuclDistTemporal}$ functions calculate the Euclidean distance between the two tweets based on their spatial and temporal components and then normalize those values to a range between zero and one based on the spatial and temporal extent of the analysis, the $\text{EuclDistSpatialTemporal}$ function in Eq. 4 calculates the spatiotemporal Euclidean distance between the two tweets using the three components of x , y , and t . In both formulas, we used the WGS 1984 Web Mercator Auxiliary Sphere projected coordinate system (EPSG: 3857) to be able to use metric units, and the time component was presented as integer number represented in seconds.

The TextualSim function in Eqs. 3 and 4 calculates the textual similarity among the texts of the two input tweets using a cosine similarity function. However, in NLP, in order to be able to apply a cosine similarity function to two textual contents, the textual contents must first be represented as numerical vectors.

2.2.3 Vectorization and Embedding of Tweets

Word2vec, GloVe, and FastText are unsupervised learning algorithms for creating vector representation of words. FastText and Word2Vec employ a neural network to train the model using a large corpus of words while GloVe uses a log-bilinear regression model for unsupervised learning of word representations.

Word2vec, developed at Google by Mikolov et al. (2013), first trains a shallow, two-tier neural network, that tries to predict the probability of a given word from its neighboring words (Continuous Bag of words—CBOW) or

guess the neighboring words of a particular word, called the word's context, given that word (Skip-Gram) using a textual corpus. Then, the hidden layer of the trained neural network is used as the embedding layer to transfer a word to its numerical feature vector counterpart while preserving the linear regularities and semantics of the underlying language. GloVe was proposed afterward as an extension to Word2vec, to consider not just the local (the neighborhoods of the words), but also the global statistical information of the words (Pennington et al. 2014). GloVe optimizes a model so that the similarity among two words is calculated through an equation in which the dot product of the numerical vectors of the words equals the log of the number of times the two words have occurred near each other in the corpus. Finally, FastText as another extension of Word2vec was proposed by Facebook (Bojanowski et al. 2017) and incorporated sub-word information by splitting words into n-grams of characters. This way, FastText can transfer any arbitrary, out-of-the-dictionary words into their vectorized counterpart.

In this study, for each tweet, $t \in P$, its vector representation, $t.v$, is calculated using the three above-mentioned word embedding methods along with two frequency-based vectorization methods of TFIDF and CV, so that $t.v = f(t.c)$. Three pre-trained models that were trained based on huge datasets from Google News, Twitter, and Wikipedia were used for Word2vec, GloVe, and FastText, respectively (Table 1). Having these models, the average vector of the vectorized representations of every word in each tweet was considered as the vectorized representation of the tweet.

Using the vectorized representation of each tweet, the similarities among tweets were calculated through Eq. 5.

$$\text{sim}(t_1, t_2) = \cos(\theta) = \frac{t_1.v \cdot t_2.v}{\|t_1.v\| \|t_2.v\|} \quad (5)$$

2.3 Backward Linking of Clusters

In order to connect the clusters that have been detected in the current iteration with the clusters of the previous iteration, a relation strength parameter is calculated for each pair of clusters in step i and step $i - 1$, using Eq. 6, where $|\cap(C_i, C_{i-1})|$ is the number of common tweets in the two clusters and $|\cup(C_i, C_{i-1})|$ is the total number of tweets in the two clusters.

$$\text{RelationStrength}(C_i, C_{i-1}) = \frac{|\cap(C_i, C_{i-1})|}{|\cup(C_i, C_{i-1})|} \quad (6)$$

For each pair of clusters, the relation strength is calculated and then each cluster in step i will be connected with the cluster in step $i - 1$ with the highest relation strength if the relation strength is higher than a

Table 1 The specification of the pre-trained models that were used for the Word2vec, GloVe, and FastText methods

	Provider	Description of the corpus	URL
WORD2VEC	Google	A corpus of about 100 billion words from Google News	https://code.google.com/archive/p/word2vec/
GLOVE	The Stanford NLP Group	A corpus of 2 billion tweets	https://nlp.stanford.edu/projects/glove/
FASTTEXT	Facebook	1 million word vectors trained on Wikipedia 2017	https://fasttext.cc/

Table 2 Comparison of metrics

Method	Coefficients	Silhouette	No. of clusters
Weighted sum	$\alpha = 0.3, \beta = 0.2, \gamma = 0.5$	0.53	24
Multiplication	–	0.27748187	20

threshold that is calculated based on the number of common tweets between the two steps.

2.4 Topic Extraction Using the Hierarchical Dirichlet Process (HDP)

Having the clusters detected at each iteration, the topic of each cluster is detected based on the text of the tweets of that cluster. In order to extract topics, an accepted approach by previous studies (Cheng and Wicks 2014; Morchid et al. 2015; Steiger et al. 2015; Capdevila et al. 2017) is to use Latent Dirichlet Allocation (LDA). However, the main problem with the utilization of LDA is the requirement of the algorithm for specifying the number of topics. Considering the dynamic and time-dependent nature of tweets, there is no proper solution for calculating the number of topics in every iteration. It should be noted that there is no significant relationship between the number of clusters and the number of topics. In order to address this problem, a new topic extraction algorithm, called Hierarchical Dirichlet Process (HDP) (Teh et al. 2006) was used in this study that, in contrast to LDA, does not need any prior information about the expected number of topics. In this study, at each iteration, HDP is trained using the whole range of tweets in that iteration, and then, the trained model is used to extract the topics for each cluster.

2.5 Evaluation Measure

The selection of proper measures for the evaluation of clustering algorithms depends on the available information and utilized methods (Guerra et al. 2012; Mary et al. 2015). Two types of evaluation measures have been used in the literature: internal indices and external indices. While external indices compare the results with the ground truth,

Table 3 Silhouette coefficient of various methods

Iteration	FastText	GloVe	Word2vec	CV	TFIDF
1	0.577	0.563	0.538	0.433	0.477
2	0.590	0.599	0.466	0.500	0.490
3	0.553	0.602	0.546	0.450	0.462
4	0.506	0.575	0.549	0.444	0.457
5	0.518	0.430	0.519	0.413	0.419
6	0.599	0.574	0.532	0.447	0.461
7	0.605	0.580	0.567	0.468	0.483
8	0.539	0.579	0.476	0.483	0.495
9	0.513	0.577	0.541	0.473	0.471
10	0.543	0.444	0.528	0.418	0.435
11	0.410	0.596	0.495	0.426	0.457
12	0.525	0.588	0.541	0.442	0.449
13	0.422	0.554	0.468	0.487	0.454
14	0.528	0.591	0.462	0.462	0.467
Average	0.531	0.561	0.516	0.453	0.463

internal indices compare the results of different algorithms to show which algorithm performs better. Using internal evaluation criteria, the output clusters with high intra-similarity and low inter-similarity get higher scores. Because it is very hard to collect ground-truth data for events that are already happening in the real world, the internal measure of the Silhouette coefficient (Rousseeuw 1987) was used in this study (Eq. 7) to compare the results of the proposed clustering algorithms with the results of DBSCAN as the base algorithm. It ranges from -1 to $+1$, where a high value indicates that the object is well matched to its cluster and poorly matched to neighboring clusters.

$$S(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))v} \quad (7)$$

In Eq. 7, $b(i)$ is the distance between an object and the nearest cluster that the object does not belong to, and $a(i)$ is the mean intra-cluster distance of an object.

3 Results

In order to test the feasibility of the proposed method, the geotagged tweets of the case study were fed to the prototype system. The system iteratively extracted events, post-processed the outputs, and visualized the results using sliding time windows with a length of 24 h ($l = 24$ h), while each time window had 12 h overlap with the previous time window ($\delta t = 12$ h). The length of the time windows was selected by iterating over time windows of 3, 6, 12, 24, and 36 h, where 24-h time windows returned a slightly better Silhouette coefficient. Therefore, the first iteration processed the tweets that were collected between 00:00 on 12 September and 00:00 on 13 September, and the

last iteration (iteration number 14) processed the tweets that were collected between 12:00 on 18 September and 12:00 on 19 September.

3.1 Parameter Selection

The best distance metric for DSTTM was selected by running the model on a subset of the dataset and comparing the output Silhouette coefficient and the number of clusters. The weighted sum metric (Eq. 3) with different combinations of alpha, beta, and gamma parameters was compared with the multiplication metric (Eq. 4). Table 2 shows that the weighted sum metric with $\alpha = 0.3$, $\beta = 0.2$, $\gamma = 0.5$ provided the best Silhouette coefficient. In addition to Silhouette, the selected weighted sum metric extracted more clusters that were denser in comparison to the multiplication metric.

3.2 Silhouette Coefficient

Table 3 presents the Silhouette coefficient of DSTTM in various iterations while using different vectorization and text embedding methods with the selected weighted sum metric. In most iterations, GloVe obtained the highest Silhouette coefficient, with an average of 0.561, while CV and TFIDF had the lowest coefficient. Although there are slight differences between the Silhouette coefficients of GloVe, FastText, and Word2vec, GloVe had the highest average.

Table 4 Total number of extracted tweet clusters

Method	FastText	GloVe	Word2vec	CV	TFIDF
Total clusters	288	302	255	233	229

Fig. 2 Examples of tweet clusters with varied densities during Hurricane Florence in North and South Carolina (on each map, tweet clusters are highlighted with different colors)

Fig. 3 Spatiotemporal distribution of tweet clusters related to Hurricane Florence in North and South Carolina

3.3 Number of Extracted Tweet Clusters

The total number of clusters extracted by DSTTM, using each text embedding method, is presented in Table 4. CV and TFIDF extracted the lowest number of clusters, while GloVe found the highest number of clusters in comparison to other methods.

4 Discussion

This section discusses how the proposed algorithm overcomes the intrinsic spatial heterogeneity of geotagged tweets, how clusters emerge and disappear over time and space, and how textual similarity techniques affect the

Table 5 List of keywords used to select important tweet clusters

Hurricane florence	Forecast	Beach	Damage
Hurricane	Today	Alert	Flood
Tornado	Tonight	Condition	Tropical
Florence	Heavy	Weather	Accident
Tstorm (thunderstorm)	Warning	Traffic	Rainbow
Safe	Shower	Flooding	Tropical
Rest	Cloudy	Rain	Physician
Storm	Come	Wind	

Table 6 Extracted topics by the hierarchical Dirichlet process (HDP) for traffic-related events during Hurricane Florence in North and South Carolina

Cluster no.	HDP topic keywords
14	Accident state eighth traffic greenville everyone great
2	Traffic accident Raleigh exit near menchie wakecounty

The implemented HDP algorithm extracted 10 keywords for each topic. In this table, some unrelated keywords were manually removed

clustering results. Finally, the results of DSTTM using OPTICS are compared to those of DBSCAN.

4.1 Heterogeneity of Geotagged Tweets

Figure 2 shows how the utilization of OPTICS enabled DSTTM to extract clusters with different densities. Considering the unique characteristic of OPTICS, DSTTM was able to address the heterogeneity in the input dataset and find clusters with different densities at different iterations. Cluster number 104 is highly dense, while the points in clusters number 23 and 99 are located far from each other. Extracting such clusters with various densities, especially during disasters, leads to the detection of significant events at both regional and local levels.

4.2 Spatiotemporal Tweet Clustering: How the Clusters Emerge and Disappear

The proposed method was able to extract clusters that were associated with Hurricane Florence. By analyzing the word cloud of the clusters that were linked together in

Fig. 4 Example of the detected clusters related to “traffic” during Hurricane Florence in North and South Carolina

Fig. 5 An example of extracted tweet clusters by **a** FastText, **b** GloVe, **c** Word2vec, **d** CV and **e** TFIDF related to Hurricane Florence in North and South Carolina (on each map, tweet clusters are highlighted with different colors)

consecutive iterations, different words related to various phases of the hurricane were identified. The results show that in the first iterations, before the storm, most clusters had keywords like “storm” and “forecast” in their word clouds, indicating that the users were discussing an upcoming storm. Monitoring and investigating the location of those clusters can provide the possibility to measure the preparedness of different areas for the coming hurricane. In contrast, the clusters that were detected after the hurricane included keywords such as “restoration,” “damage,” and “health.” Considering these keywords and the location of the respective clusters, the damaged places that needed to be considered for rescue operations could be detected.

Figure 3 presents the changes in the distribution of the clusters related to Hurricane Florence over time, where each sub-figure manifests a distinctive period. The related clusters were filtered from the list of every cluster extracted by the application using the keywords in Table 5. The number of clusters related to the hurricane increased over time and peaked on 14 September when the hurricane made landfall on the beaches of North Carolina. As time passed, the number of hurricane-related tweet clusters gradually decreased until 18 September, when the minimum number of clusters was observed.

Figure 3 also shows the way the clusters emerged. In the early stages of the hurricane landfall, most of the clusters related to the hurricane appeared near the coastline, but they moved from the beaches, inward, to the west of North Carolina over time. These clusters mostly include keywords such as “Hurricane,” “Florence,” “Tornado,” “Storm,” “Flood,” “Rain,” “Shower,” “Wind,” and “Cloudy.” The spatiotemporal clusters extracted over time were following the path of the hurricane. In large-scale disasters like Hurricane Florence, where many victims need assistance, detecting the places that are severely affected by the disaster along the path of the event is highly valuable and can help disaster managers allocate their resources better.

Two noticeable clusters detected by the system during the hurricane were the ones related to traffic and accidents

(Table 6, Fig. 4), one in North Carolina, and the other in South Carolina. The cluster in South Carolina (Fig. 4b) lasted for 2 days, contained “accident” and “traffic” keywords, and appeared because of an accident in the area. The cluster in North Carolina (Fig. 4a) existed from the beginning of the analysis to the last day, contained the keywords “traffic” and “accident,” emerged near Raleigh and Durham cities and presented permanent traffic in this area.

4.3 The Effect of Vectorization and Text Embedding Methods

Comparing the output clusters resulting from utilization of FastText, GloVe, Word2vec, TFIDF, and CV shows that TFIDF and CV extracted similar clusters while FastText, GloVe, and Word2vec had almost the same behavior in cluster extraction. The difference between the two groups is related to the size and the number of clusters, as well as the distribution of tweets in clusters. Fewer clusters are extracted by TFIDF and CV, and the extracted clusters are larger than those extracted by FastText, Word2vec, and GloVe. Examples of clusters extracted by each method are illustrated in Fig. 5 where the distribution, number, and size of clusters can be compared.

Moreover, TFIDF and CV extracted some clusters in which points are distributed over the whole study area. Extracted clusters by TFIDF and CV were larger than those extracted by FastText, GloVe, and Word2vec. It was observed that big clusters extracted by TFIDF and CV were broken into smaller clusters with more details when applying FastText, GloVe, and word2vec for textual similarity. This means that TFIDF and CV could not efficiently separate the words related to different topics. GloVe, in comparison to FastText and Word2vec, could extract clusters with more details in some cases. In Fig. 6, clusters 148 and 152 extracted by FastText and Word2vec, for example, were broken into two smaller clusters by GloVe (140 and 191) with more details. Checking the word clouds of extracted clusters and their topics (Table 7) shows that

Fig. 6 An example of two tweet clusters extracted by **a** GloVe, compared to one cluster by **b** FastText and **c** Word2vec related to Hurricane Florence in North and South Carolina

GloVe extracted two clusters with words “Raleigh, traffic, accident” and “Durham, traffic, flooding,” pointing to traffic in two different cities with dissimilar causes. FastText and Word2vec combined these two clusters because both clusters were related to the same traffic event.

4.4 Comparing the Results of Dynamic Spatio-Temporal Tweet Mining (DSTTM) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN, as the commonly used algorithm for event detection from Twitter data, was chosen as a base algorithm to be compared with DSTTM. Since the selection of input parameters of DBSCAN could significantly influence the output result, we used the K-dist plot to determine the epsilon parameter for DBSCAN.

Figure 7 presents the output of DSTTM in comparison to DBSCAN. The figure shows that DBSCAN extracted clusters with almost the same densities in relation to the epsilon value that was computed from K-dist plots, and neglected the clusters with varied densities. In comparison, DSTTM extracted clusters with different densities and was able to extract local clusters with more details in comparison with DBSCAN. Figure 8a, b, for example, show the same clusters extracted by DSTTM and DBSCAN, respectively. As word clouds show, DSTTM was able to divide one cluster extracted by DBSCAN into two separate clusters with different sets of words, including “Hurricane Florence” and “Traffic accident.” Having separated clusters with more details will help managers and decision makers to accurately locate each event and set the required measures to deal with each situation appropriately.

5 Conclusion

This study proposed DSTTM as a method for dynamic spatiotemporal event extraction from Twitter data that can be used in large study areas for disaster management purposes. DSTTM was implemented and tested through a case study related to Hurricane Florence. Analyzing the content, location, and time of extracted clusters proved that the proposed method can detect clusters with varied sizes and densities in the course of events that affect large study areas. The real-time information, extracted by DSTTM, can be used by decision makers and disaster managers for rapid and effective responses to different incidents before, during, and after a disaster.

As future work, we will extend DSTTM and utilize new clustering approaches that can directly deal with the high-dimensional space of the embedded texts along with the spatial and temporal components. In this regard, we

will chiefly concentrate on the exploitation of soft subspace clustering algorithms as well as multiview clustering methods and compare their performance with the density-based algorithms. Moreover, the effect of spatial autocorrelation among tweets on the event detection from geotagged tweets will be analyzed further. We will also try to apply DSTTM to other types of disasters that affect large study areas and assess its performance and feasibility.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Arcaïni, P., G. Bordogna, D. Ienco, and S. Sterlacchini. 2016. User-driven geo-temporal density-based exploration of periodic and not periodic events reported in social networks. *Information Sciences* 340–341: 122–143.
- Benhardus, J., and J. Kalita. 2013. Streaming trend detection in twitter. *International Journal of Web Based Communities* 9(1): 122–139.
- Ben-Lhachemi, N., and E.H. Nfaoui. 2018. Using tweets embeddings for hashtag recommendation in twitter. *Procedia Computer Science* 127: 7–15.
- Bifet, A. 2010. *Adaptive stream mining: Pattern learning and mining from evolving data streams*. Amsterdam: IOS Press.
- Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5: 135–146.
- Capdevila, J., J. Cerquides, J. Nin, and J. Torres. 2017. Tweet-SCAN: An event discovery technique for geo-located tweets. *Pattern Recognition Letters* 93: 58–68.
- Cheng, T., and T. Wicks. 2014. Event detection using twitter: A spatio-temporal approach. *PloS One* 9(6): Article e97807.
- Croitoru, A., N. Wayant, A. Crooks, J. Radzikowski, and A. Stefanidis. 2015. Linking cyber and physical spaces through community detection and clustering in social media feeds. *Computers, Environment and Urban Systems* 53: 47–64.
- Cui, W., P. Wang, Y. Du, X. Chen, D. Guo, J. Li, and Y. Zhou. 2017. An algorithm for event detection based on social media data. *Neurocomputing* 254: 53–58.
- Ester, M., H.-P. Kriegel, J. Sander, and X. Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the international conference on knowledge discovery and data mining*, 226–231, 2–4 August 1996, Portland, OR, USA.
- Farnaghi, M., and A. Mansourian. 2013. Disaster planning using automated composition of semantic OGC web services: A case study in sheltering. *Computers, Environment and Urban Systems* 41: 204–218.

- Fócil-Arias, C., J. Zúñiga, G. Sidorov, I. Batyrshin, and A. Gelbukh. 2017. A tweets classifier based on cosine similarity. Working notes of CLEF 2017—Conference and Labs of the Evaluation Forum, Dublin, Ireland, 11–14 September 2017.
- Ghaemi, Z., and M. Farnaghi. 2019. A varied density-based clustering approach for event detection from heterogeneous twitter data. *ISPRS International Journal of Geo-Information* 8(2): Article 82.
- Guerra, L., V. Robles, C. Bielza, and P. Larrañaga. 2012. A comparison of clustering quality indices using outliers and noise. *Intelligent Data Analysis* 16(4): 703–715.
- Hasan, M., M.A. Orgun, and R. Schwitter. 2018. A survey on real-time event detection from the Twitter data stream. *Journal of Information Science* 44(4): 443–463.
- Hecht, B., L. Hong, B. Suh, and E.H. Chi. 2011. Tweets from Justin Bieber's heart: The dynamics of the "location" field in user profiles. In *Proceedings of the ACM CHI annual conference on human factors in computing systems*, 237–246, 7–12 May 2011, Vancouver, BC, Canada.
- Huang, Q., and Y. Xiao. 2015. Geographic situational awareness: Mining tweets for disaster preparedness, emergency response, impact, and recovery. *ISPRS International Journal of Geo-Information* 4(3): 1549–1568.
- Huang, Y., Y. Li, and J. Shan. 2018. Spatial-temporal event detection from geotagged tweets. *ISPRS International Journal of Geo-Information* 7(4): Article 150.
- Idrissi, A., H. Rehioui, A. Laghrissi, and S. Retal. 2015. An improvement of DENCLUE algorithm for the data clustering. In *Proceedings of the 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA)*, 21–23 December 2015, Marrakech, Morocco. IEEE. <https://doi.org/10.1109/icta.2015.7426936>.
- Joshi, A., and R. Kaur. 2013. A review: Comparative study of various clustering techniques in data mining. *International Journal of Advanced Research in Computer Science and Software Engineering* 3(3): 55–57.
- Kaleel, S.B., and A. Abhari. 2015. Cluster-discovery of twitter messages for event detection and trending. *Journal of Computational Science* 6: 47–57.
- Kirilenko, A.P., and S.O. Stepchenkova. 2017. Sochi 2014 Olympics on twitter: Perspectives of hosts and guests. *Tourism Management* 63: 54–65.
- Krajewski, W.F., D. Ceynar, I. Demir, R. Goska, A. Kruger, C. Langel, R. Mantilla, J. Niemeier, et al. 2016. Real-time flood forecasting and information system for the State of Iowa. *Bulletin of the American Meteorological Society* 98(3): 539–554.
- Lee, C.-H. 2012. Mining spatio-temporal information on microblogging streams using a density-based online clustering method. *Expert Systems with Applications* 39(10): 9623–9641.
- Lee, K., D. Palsetia, R. Narayanan, M.M.A. Patwary, A. Agrawal, and A.N. Choudhary. 2011. Twitter trending topic classification. In *Proceedings of the 11th IEEE international conference on data mining workshops*, 251–258, 11 December 2011, Vancouver, BC, Canada.
- Liu, P., D. Zhou, and N. Wu. 2007. VDBSCAN: Varied density based spatial clustering of applications with noise. In *Proceedings of the 2007 international conference on service systems and service management*, 1–4, 9–11 June 2007, Chengdu, China.
- Mary, S.A.L., A.N. Sivagami, and M.U. Rani. 2015. Cluster validity measures dynamic clustering algorithms. *ARPN Journal of Engineering and Applied Sciences* 10(9): 4009–4012.
- Mikolov, T., K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the 1st international conference on learning representations*, 1–12, 2–4 May 2013, Scottsdale, AZ, USA.
- Morchid, M., Y. Portilla, D. Josselin, R. Dufour, E. Altman, M. El-Beze, J.-V. Cossu, G. Linarès, and A. Reiffers-Masson. 2015. An author-topic based approach to cluster tweets and mine their location. *Procedia Environmental Sciences* 27: 26–29.
- Nguyen, M.D., and W.-Y. Shin. 2017. DBSTexC: Density-based spatio-textual clustering on twitter. In *Proceedings of the 9th IEEE/ACM international conference on advances in social networks analysis and mining*, 23–26, 31 July–3 August 2017, Sydney, Australia.
- Nguyen, T., M.E. Larsen, B. O'Dea, D.T. Nguyen, J. Yearwood, D. Phung, S. Venkatesh, and H. Christensen. 2017. Kernel-based features for predicting population health indices from geocoded social media data. *Decision Support Systems* 102: 22–31.
- Niederkrötenhaler, T., B. Till, and D. Garcia. 2019. Celebrity suicide on twitter: Activity, content and network analysis related to the death of Swedish DJ Tim Bergling alias Avicii. *Journal of Affective Disorders* 245: 848–855.
- Parimala, M., D. Lopez, and N.C. Senthilkumar. 2011. A survey on density based clustering algorithms for mining large spatial databases. *International Journal of Advanced Science and Technology* 31(1): 59–66.
- Pennington, J., R. Socher, and C. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543, 25–29 October 2014, Doha, Qatar.
- Phelan, O., K. McCarthy, and B. Smyth. 2009. Using twitter to recommend real-time topical news. In *Proceedings of the 2009 ACM conference on recommender systems*, 385–388, 23–25 October 2009, New York, NY, USA.
- Reddy, B.G.O., and M. Ussenaiah. 2012. Literature survey on clustering techniques. *IOSR Journal of Computer Engineering* 3(1): 1–50.
- Rousseeuw, P.J. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20: 53–65.
- Sander, J., M. Ester, H.-P. Kriegel, and X. Xu. 1998. Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery* 2(2): 169–194.
- Schubert, E., and M. Gertz. 2018. Improving the cluster structure extracted from OPTICS plots. In *Proceedings of the conference "lernen, wissen, daten, analysen"*, 318–329, 22–24 August 2018, Mannheim, Germany.
- Schubert, E., J. Sander, M. Ester, H.P. Kriegel, and X. Xu. 2017. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems* 42(3): Article 19.
- Sit, M.A., C. Koylu, and I. Demir. 2019. Identifying disaster-related tweets and their semantic, spatial and temporal context using deep learning, natural language processing and spatial analysis: A case study of Hurricane Irma. *International Journal of Digital Earth* 12(11): 1205–1229.
- Srijith, P.K., M. Hepple, K. Bontcheva, and D. Preotiuc-Pietro. 2017. Sub-story detection in twitter with hierarchical Dirichlet processes. *Information Processing & Management* 53(4): 989–1003.
- Steiger, E., B. Resch, and A. Zipf. 2016. Exploration of spatiotemporal and semantic clusters of Twitter data using unsupervised neural networks. *International Journal of Geographical Information Science* 30(9): 1694–1716.
- Steiger, E., R. Westerholt, B. Resch, and A. Zipf. 2015. Twitter as an indicator for whereabouts of people? Correlating twitter with UK census data. *Computers, Environment and Urban Systems* 54: 255–265.
- Sutton, J., S.C. Vos, M.K. Olson, C. Woods, E. Cohen, C.B. Gibson, N.E. Phillips, J.L. Studts, et al. 2018. Lung cancer messages on

- twitter: Content analysis and evaluation. *Journal of the American College of Radiology* 15(1): 210–217.
- Teh, Y.W., M.I. Jordan, M.J. Beal, and D.M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association* 101(476): 1566–1581.
- Vijayarani, S., and P. Jothi. 2014. Partitioning clustering algorithms for data stream outlier detection. *International Journal of Innovative Research in Computer and Communication Engineering* 2(4): 3975–3981.
- Walther, M., and M. Kaisser. 2013. Geo-spatial event detection in the twitter stream. In *Proceedings of the 35th European conference on advances in information retrieval, ECIR 2013*, 356–367, 24–27 March 2013, Moscow, Russia.
- Wang, Z., X. Ye, and M.-H. Tsou. 2016. Spatial, temporal, and content analysis of Twitter for wildfire hazards. *Natural Hazards* 83(1): 523–540.
- Yang, W., and L. Mu. 2015. GIS analysis of depression among twitter users. *Applied Geography* 60: 217–223.