IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 30, 2020, accepted June 28, 2020, date of publication July 13, 2020, date of current version July 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3008641

Automatic Event Geo-Location in Twitter

GIOVANNI ACAMPORA[1], (Senior Member, IEEE), PAOLO ANASTASIO[2], MICHELE RISI[3], (Member, IEEE), GENOVEFFA TORTORA[3], (Senior Member, IEEE), AND AUTILIA VITIELLO[1], (Member, IEEE)

[1]Department of Physics "Ettore Pancini", University of Naples Federico II, 80126 Naples, Italy
[2]Spike Reply, 20141 Milan, Italy
[3]Department of Computer Science, University of Salerno, 84084 Fisciano, Italy

Corresponding author: Autilia Vitiello (autilia.vitiello@unina.it)

ABSTRACT Twitter is currently one of the most popular platforms for disseminating information about events happening around the world. Especially but not only for emergency events, it is crucial to know when and where the events are taking place. Unfortunately, identifying the geo-location of an event discussed in Twitter is a very challenging task mainly due to the brevity of the messages (i.e., tweets) and their subjective nature. In the literature, some efforts have been made to address this task, but they are characterized by substantial limitations such as the use of exclusively text analysis techniques, or the need for keywords or possible candidate locations. This paper proposes a new process for automatic event geo-localization which relies on both textual and spatial/temporal use of content posted on Twitter without using some prior knowledge about the event to be located. As shown by experimental results, our proposal achieves a good accuracy rate and outperforms two well-known baseline approaches related to the geo-location of events in Twitter.

INDEX TERMS Big data, data-mining, event-localization, Twitter.

I. INTRODUCTION

Microblogging is a broadcast medium that allows users to create small digital content such as short texts, links, images, or videos and share it with an online audience. Although this communication medium is relatively new with respect to traditional media, it has gained increased attention among online users thanks to its immediacy and portability. Indeed, online users can instantly respond and spread information by using a variety of computing devices, including smartphones and tablets. Among the existing microblogging services, Twitter is currently the most popular one with 330 million monthly active users as of the fourth quarter of 2017. In particular, Twitter enables users to post text messages, known as tweets, no longer than 280-characters (until November 2017, this limit was 140-characters) into a digital space, known as the Twittersphere [1]. People use Twitter to express their feelings, thoughts, and comments about an event that they have witnessed or heard about [2]. In several studies, an event is defined as a unique thing that happens *at a specific time* [3], attracting people's attention, thus generating a message traffic [2]. Therefore, monitoring and analyzing Twitter streams can enable individuals, corporations, and government organizations to stay informed of *"what is happening now"* [4]. Hence, there is a fervent field of research aimed at developing event detection tools from Twitter streams. However, in several scenarios, it would be useful to enrich the information "what is happening now" with the information *"where it is happening"*. As a consequence, in this work, an event is defined as a unique thing that happens *at a specific time and place*. Unfortunately, identifying the location where events are taking place is one of the biggest challenges in this new field of research. The complexity of event geo-localization is related to a set of factors which include: *a)* not all tweets written during the discussion of an event contain information about the location of that event; *b)* only a very small part of the tweets (about 2% of the posted tweets [5]) are geo-located, indeed, not necessarily a user in describing an event must also indicate the place where it occurred; *c)* the information contained in the tweets may be inconsistent (e.g., inaccurate, badly written, ambiguous). Because of its underlying complexity, in the literature, only few efforts, characterized by crucial limitations, have been carried out to address the geo-localization problem. The main limitations of these approaches can be resumed in: *i)* using keywords that reduce the space of the search; *ii)* relying

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Marozzo.

on information often unknown such as a set of candidate locations; *iii)* getting locations through an exclusively textual analysis of the tweets. Starting from these considerations, this work proposes a new approach to geo-localize an event discussed in Twitter which relies on both textual and spatial/temporal use of content posted on Twitter without using some prior knowledge about the event to be located. Broadly, the proposed process exploits 1) text analysis techniques and clustering for identifying tweets related to the same event; 2) a time frequency analysis to identify the most significant tweets related to a given event 3) data cleaning techniques based on vocabularies and Google Maps API for defining the area in which to geo-localize an event.

The proposed approach can be used to identify places in the world where events such as concerts, festivals, sports and disasters have been occurring. Hence, no limitation on the space window or the kind of event is considered. However, it is worth noting that the proposed approach is an offline method to extract knowledge. Hence, it starts after collecting a set of tweets in an any time period of interest and achieves the information about the localization of the events in the considered time period. Therefore, it is fair to say that the proposed approach is not suitable to be used in situations where real-time features are required such as identification of disaster locations for supporting emergency management.

In order to evaluate the proposed process, three datasets were collected, respectively, in three different days between April 17 and June 6, 2016. As shown by the results of our experiments, the proposed process successfully geo-locates approximately 80% of events. This result shows the suitability of our proposal in the automatic geo-localization of events in Twitter.

The remaining of the paper is as follows. Section II discusses state-of-the-art approaches in the literature. Section III describes our proposal to geo-locate events in Twitter. The settings and results of the carried out experiments are reported in Section IV. Conclusion and future work are outlined in Section V.

II. RELATED WORK

In the literature, several works have been developed to manage Twitter data [6], [7], but the most part of them focuses on solving the event detection problem [1], [4], [8]–[11] or the geo-localization of users [12]–[18], whereas, to the best of our knowledge, only recently researches are devoted to geo-locate Twitter events [2].

The first efforts in this field are devoted to geo-localize emergency events [19]. For example, Ao *et al.* [20] propose a system to estimate the location of emergency events from Twitter-like data captured from Sina Weibo, the most popular microblogging service in China. The system takes in input a keyword to characterize an event such as an earthquake or a tsunami and computes the location of this event by using information about the content of weibos, the Global Positioning System (GPS) location of a weibo and the registration location of the user who has written the weibo. Also Giridhar *et al.* [21] propose a system to localize a Twitter event starting from a keyword for the event given by the user. In detail, this system identifies distinct event signatures in the blogosphere, clusters microblogs based on events they describe, and analyzes the resulting clusters to extract the locations. This information is then translated using the Google Maps API for geo-location, offering a real-time view of ongoing events on a map. Both the systems presented in [20] and [21] surely share with our approach the final goal of computing the location of an event, but they are characterized by a different starting point. Indeed, these systems require the user to specify a keyword, i.e., the topic of the event to be geo-located. However, this approach can reduce the search space by removing useful information to better geo-locate the event. On the contrary, our method is able to geo-locate an event by considering all the related tweets and without the need to specify a keyword or topic.

Among the other literature approaches, the work by Ozdikis *et al.* [22] deserves to be discussed. This work applies for the first time the Demspter-Shafer Theory (DST) for this problem. In detail, given a set of tweets and a set of cities, the proposed approach aims to extract evidence in tweets and generate belief intervals for cities where the event might have happened. However, also for this system, the starting point is very different from our approach. Indeed, this system works by having in input a set of possible cities, instead, our approach computes the location of a Twitter event from scratch. Another interesting work [23] presents the tool TwitterTagger. This tool is capable of geo-tagging tweets by using their textual content. In particular, the tool tags the content of each tweet and, then performs two disambiguations in order to clarify the connotations of the noun phrases in each tweet and to associate correct locations with each tweet. The final goal of this approach is to show relevant tweets to a user based on his/her physical location. Even if this approach is related to geo-location in Twitter, it is characterized by a different goal with respect to our method. Indeed, TwitterTagger tries to geo-tag each single tweet to detect which are those relevant for a user, instead, our approach is devoted to geo-localize a set of related tweets in order to provide users with the geo-location of events discussed in Twitter.

Finally, Khanwalkar *et al.* [24] present a content-based, geo-location detection approach that is capable of geo-locating multilingual tweets, within a time window, by exclusively using the textual content of the considered tweets. The choice of using a time window is based on the intuition that Twitter users post tweets on specific trending topics and move on to other topics within a certain temporal window. Initially, the system converts the set of tweets captured during a time-delineated window into a set of documents, where each document contains multiple tweet posts from a specific user. Then, the system manages the multilingual tweets by translating all tweets in English. Finally, the system uses a named entity detection algorithm and k-means clustering to detect a geographic location for each document. Therefore, the system identifies the location

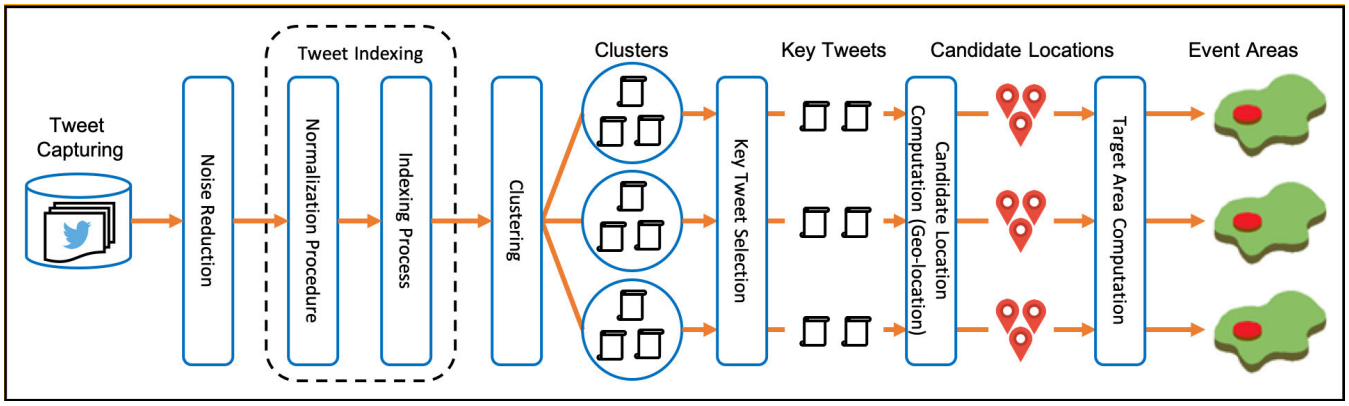

FIGURE 1. The proposed process to geo-locate events in Twitter.

related to a set of tweets contained in a document rather than the location of an event. Indeed, there is not a system component devoted to guarantee that the set of tweets in a document represents an event or that two documents represent the same event. Instead, our approach identifies the location of an event by clustering different tweets that share the same content.

Starting from this discussion, the approach proposed in this work has the important benefit of performing the geo-localization of a set of related tweets characterizing an event without using keywords and possible candidate locations.

III. OUR PROPOSAL TO THE GEO-LOCATION OF EVENTS IN TWITTER

The goal of our proposal is to compute the geo-location of the events discussed in Twitter. The proposed approach detects events in Twitter by clustering individual tweets and computes the geo-location of these events by analyzing the obtained clusters by means of the Google Maps geo-location services.

The main process underlying our approach is composed of seven phases (see Fig. 1): *tweet capturing*, *noise reduction*, *tweet indexing*, *clustering*, *key tweet selection*, *candidate location detection* and *target area computation*. Hereafter, details about each one of these phases will be discussed.

A. TWEET CAPTURING

The tweet capturing phase works by using the Streaming API,[1] provided by Twitter. In particular, this API allows to obtain contents of Twitter under the format of a streaming in real time and with a low latency. The request and the transfer of contents is through an HTTP persistent connection, whilst contents are structured by using the format JavaScript Object Notation (JSON).[2] For each tweet the following information is extracted:

- *id*: an integer number representing a unique identifier for the tweet;
- *text*: the text of the tweet in UTF-8 encoding;
- *iso_language_code*: the language of the tweet in standard ISO 639-1;
- *created_at*: Coordinated Universal Time (UTC) when the tweet was created;
- *place.name*: the name of a location which the tweet is associated to by the user when the tweet is posted. Being set by the user, this attribute is not always present.

It is worth noting that the proposed approach does not exploit all geo-spatial tweet metadata. In particular, as well as the tweet location (the attribute *place.name*), geo-located information is also contained in the user's location attribute (set by users during the creation of their profile). Some works [20], [22] take into account the user's location due to the assumption that users tend to post tweets about events near to their location [2]. However, to be honest, users can discuss about everything located everywhere, therefore, user's location could be misleading in the event location detection. Hence, the proposed approach does not exploit this geo-spatial information.

B. NOISE REDUCTION

In order to reduce the amount of data and improve the performance, the goal of the noise reduction phase is to remove tweets that can provide wrong information. In particular, the factors taken into account to remove these tweets are:

- *the number of words*: the tweets composed of only one word are deleted because it is highly unlikely that they can contain useful information;
- *the retweets*: the retweets are deleted because they do not add further information with respect to those already posted by other users. Retweets are easily identifiable because they start with the word "RT";
- *Public messages*: public messages for a specific user are deleted because they are related to personal conversations and, as a consequence, it is highly unlikely that they

[1] https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data

[2] https://www.json.org

can contain information about an event. These tweets are easily recognizable because they start with a Twitter mention.

C. TWEET INDEXING

This phase performs the indexing of the tweets provided as output by the noise reduction phase. The indexing procedure consists of representing each tweet in the Vector Space Model (VSM) [25]. Before indexing tweets, they are undergone to a normalization procedure aimed at cleaning them from non-standard words written both intentionally and unintentionally by people. Hereafter, details about the normalization and indexing procedures are given.

1) NORMALIZATION PROCEDURE

This procedure is performed during the tweet indexing phase. It consists of the following steps:

- *Removal of emojis, numbers and other non-alphanumeric symbols*: the emojis are non-alphanumeric symbols (i.e., not composed of numbers and letters) that are used in tweets (see an example in Fig. 2). These are different smiles and symbols that allow users to give a tone and emotion to the messages. Although the emojis can be an important element to describe emotions and moods about personal events and events featured on Twitter [26], we remove them together with numbers and non-alphanumeric symbols since they cannot contain useful information for the purpose of detecting and geo-localizing events.
- *Removal of the stop words*: the stop words are the most common words in a language and we remove them because of their poor influence in identifying a document uniquely. Each language is characterized by a list of different stop words including articles, conjunctions, prepositions and auxiliary verbs.
- *Removal of hyper-textual links*: each link introduced in a Twitter text message is automatically converted in a special format. Thus, they can be easily identified and removed since they cannot contain useful information for detecting and geo-localizing events.
- *Removal of mentions*: using mentions is a method to draw attention of a specific user to a message. A mention is characterized by the following format: the symbol @ followed by the user name. Therefore, a mention is not useful for detecting and geo-localizing events and, for this reason, we remove it.
- *Removal of two-characters words*: it has been chosen to remove these words because it is assumed that it is highly unlikely that they contain useful information for our purpose.

FIGURE 2. Some emojis in Twitter.

- *White Space Strip*: spaces in excess (i.e., spaces at the beginning or the ending of the tweets or multiple spaces among words) are removed.
- *Case-folding*: all the words have been converted in lower case in order to facilitate their comparison.
- *Stemming*: a stemming process to reduce words to their word stem, base or root form is used. This step is useful to shorten the vocabulary space and, as a consequence, drastically improve the indexing process. The used stemming is a consolidated approach proposed by Porter [27] in 1980. Fig. 3 shows an example for the used stemming process.

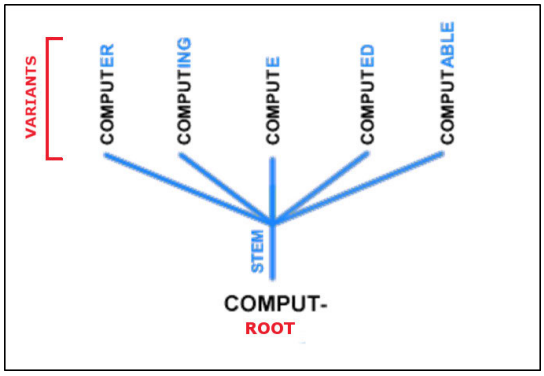

FIGURE 3. An example of the used stemming process.

After the normalization procedure, the proposed process performs the tokenization, i.e., the breaking up of the text of the tweets into terms. This task is executed by locating word boundaries delimited by white spaces. Once the texts of the tweets are divided in terms, the tweet indexing phase includes the indexing process discussed below.

2) INDEXING PROCESS

The indexing process consists of representing each tweet in the Vector Space Model (VSM), a feature space typically used in the Information Retrieval (IR) field. In detail, each tweet is stored as a vector of terms, each one characterized by a weight representing its importance in the tweet and within the whole set of tweets. Formally,

$$\vec{d}_j = (w_{1j}, w_{2j}, \ldots, w_{kj}) \tag{1}$$

where $\vec{d}_j$ is the vector of the j-th tweet, w_{ij} is the weight of the i-th term of the j-th tweet, and k is the number of the terms in the j-th tweet. The weight of a term in a tweet vector can be determined in many ways. We use one of the most typical methods in IR that has been demonstrated to be very efficient [28], the Term Frequency - Inverse Document Frequency (TF-IDF) method. This technique allows the computation of the weight of a term by using two factors: how often the term i occurs in the tweet j and how often it occurs in the whole set of tweets. Formally, the weight of a term i in tweet j is:

$$w_{ij} = tf_{ij} \times idf_i \tag{2}$$

where tf_{ij} is computed by using the term frequency f_{ij} (i.e., the number of times that the term i appears in tweet j) as follows:

$$tf_{ij} = \begin{cases} 1 + \log(f_{ij}) & \text{if } f_{ij} > 0 \\ 0 & \text{if } f_{ij} = 0 \end{cases} \quad (3)$$

and idf_i stands for the so-called inverse document frequency and is computed as follows:

$$idf_i = 1 + \log(N/df_i) \quad (4)$$

where N is the number of tweets and df_i is the number of tweets that contain the term i. Intuitively, the method assigns high weights to terms that appear frequently in a small number of tweets in the set (because they have a strong discriminant power) and low weights to terms that appear seldom in a tweet or frequently in the set of tweets (because in this case the terms are not very useful for distinguishing a tweet from others).

Once the term weights are determined, it is necessary to select a ranking function to measure similarity between each couple of tweets. A common similarity measure in IR successfully applied together with TF-IDF method is known as the cosine similarity measure. This measure is based on the computation of the angle between two tweet vectors. The cosine of 0 is 1, whereas, the cosine of 90 is 0. Since the maximum angle between two tweet vectors is 90, the cosine similarity measure ranges from 0 to 1. In detail, if the two vectors are not so similar, the angle between them is very large and the cosine tends to 0; if the vectors are very similar, the angle between them is small and the cosine tends to 1. Formally, the cosine similarity measure sim_{cos} between two tweet vectors $\vec{d}_1$ and $\vec{d}_2$ is computed as follows:

$$sim_{cos}(\vec{d}_1, \vec{d}_2) = \frac{\vec{d}_1 \cdot \vec{d}_2}{||\vec{d}_1|| \cdot ||\vec{d}_2||}. \quad (5)$$

Starting from the cosine similarity measure, it is possible to define also the cosine dissimilarity measure dis_{cos} between two tweet vectors $\vec{d}_1$ and $\vec{d}_2$ as follows:

$$dis_{cos}(\vec{d}_1, \vec{d}_2) = 1 - sim_{cos}(\vec{d}_1, \vec{d}_2) \quad (6)$$

that will be used in the clustering process described in the next section.

D. CLUSTERING

The clustering phase is devoted to divide the set of tweets in groups, where each one of them contains the tweets related to a single Twitter event. In order to achieve this aim, the clustering process executes the Partition Around Medoids (PAM) method, a partitional algorithm proposed by Leonard Kaufman and Peter J. Rousseeuw in 1987 [29]. This algorithm is similar to K-means, since both the algorithms separate data into k clusters by trying to minimize the distance between the points associated with a cluster and a point designated as the center of that cluster. However, they differ for the choice of the center of the clusters. Indeed, K-means uses centroids, whereas, PAM works with medoids. A medoid is the most centrally located point in the cluster, whereas, a centroid is a point artificially created by computing the mean distance between points in the cluster. The choice of the PAM algorithm is motivated by the fact that PAM is more robust than K-means since the exploitation of medoids is less affected than centroids by the presence of outliers. The PAM algorithm works as follows. Initially, it selects from the dataset an initial set of k medoids arbitrarily. After finding k medoids, k clusters are constructed by assigning each point to the nearest medoid. The goal is to find k medoids which minimize the sum of the dissimilarities of the points to their closest representative medoids. Then, it replaces one of the medoids with another point in the dataset if this minimizes further the sum of the dissimilarities to all the points in the cluster. This step is repeated until no change in medoids will decrease the sum of the dissimilarities. In our context, points of the dataset will be tweets. Therefore, our clustering phase uses the PAM algorithm with the following inputs 1) the dataset of n tweets to be separated, 2) the cosine dissimilarity matrix computed during the indexing process phase and 3) the value k representing the number of clusters to obtain (i.e., the number of Twitter events present in the set of tweets).

In order to obtain the best value for k, the proposed process executes the silhouette method [30]. Formally, the clustering process computes the PAM algorithm for different values of k, with $k \geq 3$, by obtaining k different separations of the set of the tweets. Then, it computes the silhouette value for each one of the k separations sil_k as follows:

$$sil_k = \frac{\sum_{i=1}^{n} s_{i,k}}{n} \quad (7)$$

where n is the number of the tweets and $s_{i,k}$ is the silhouette coefficient of the i-th tweet in the k-th separation. In turn, the value $s_{i,k}$ is obtained as follows:

$$s_{i,k} = \frac{b_{i,k} - a_{i,k}}{max(b_{i,k}, a_{i,k})} \quad (8)$$

where $b_{i,k}$ is the lowest average dissimilarity of the i-th tweet of the k-th separation to any other cluster present in the separation but not containing that tweet, whereas, $a_{i,k}$ is the average dissimilarity of the i-th tweet of the k-th separation with all other tweets within the same cluster. After computing the values sil_k for each one of the k separations, the clustering process selects the separation with the highest value. It is worth noting that, during the setting of the different values for k, the clustering process leaves out the base case $k = 2$. This choice depends on the consideration that it is highly unlikely that only two events are discussed in Twitter at the same time. Moreover, it is worth noting that the exploitation of the silhouette method allows clustering tweets collected during time windows of any length. Indeed, the silhouette method allows dividing the tweets in the most suitable number of events regardless of the number of tweets.

E. KEY-TWEET SELECTION

After computing the clustering of the set of tweets collected in the time window of interest, each cluster can contain a large number of tweets identifying an event. In this amount of tweets our approach must search information useful to obtain the geo-location of that event. In order to facilitate this task, the key tweet selection phase tries to reduce the amount of tweets related to an event by selecting the most representative ones contained in the corresponding cluster. The selection of the key tweets is based on an important observation: when an event occurs, the number of tweets widely increases [1]. Therefore, the tweets contained around a temporal peak of the Twitter activity can be considered the most relevant for that event. Starting from this consideration, the key tweet selection phase sorts the tweets contained in a cluster for tweet-arrival time and, then, selects the tweets contained in the temporal peak area as key tweets (see Fig. 4). In detail, the proposed process identifies the temporal peak area associated with a cluster of tweets by means of the exploitation of the Offline Peak-Finding Algorithm (OPFA) proposed in [31]. This algorithm bins the tweets belonging to a cluster into a histogram by time (in our case, by minutes). Then, the OPFA algorithm calculates a historically weighted running average of tweet rate and identifies rates that are significantly higher than the mean tweet rate [31]. For these rate spikes, the OPFA algorithm finds the local maximum of tweet rate and identifies a window surrounding the local maximum. The tweets contained in this window are considered to be the most representative ones.

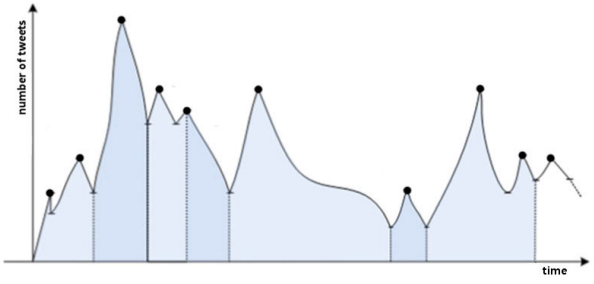

FIGURE 4. An example of the temporal peaks identified by the OPFA algorithm.

F. CANDIDATE LOCATION COMPUTATION

Starting from the most important tweets related to each event, during the candidate location computation phase, the proposed process extracts a set of candidate locations for the event. In detail, during this phase, a textual analysis of the key tweets is performed by removing all terms that are not related to a location with high probability. In order to achieve this aim, the tweet terms are compared with the words contained in a set of dictionaries: if the tweet term is found in one of the used dictionaries, it is removed. The used dictionaries are: 1) the dictionary of the language of the tweet[3]; 2) the English dictionary[4] because the use of anglicisms is a common routine; 3) a dictionary of the Internet slang[5] containing the most common abbreviations used in the Internet. The tweet terms that are not filtered are considered to be names for candidate locations. Moreover, since the name of a location could be composed of more than one word, other candidate location names are created by combining the unfiltered tweet terms that are consecutive. The set of candidate location names obtained by the textual analysis of the key tweets is enriched with the location names indicated by the users during the publication of the key tweets (i.e., JSON field *place.name*). However, unfortunately, this information that is very useful is seldom provided by the users, as highlighted in [32]. Therefore, in the most part of the cases, the candidate location names will be originated only by the textual analysis of the key tweets.

Once computed the set of candidate location names, it uses the Google Maps API[6] to verify whether the candidate location names correspond to effective locations. In particular, for each candidate location name, the Google Maps API verifies whether there exists an exact, partial or null match with real locations. In this way, only the geographic coordinates of the locations with an exact match with one of the candidate location names are stored.

G. TARGET AREA COMPUTATION

Once the candidate location computation phase is ended, the resulting set of locations (with their geographic coordinates) is given in input to the target area computation phase. During this phase, these locations are analyzed to identify the geographic area where the event is occurred (i.e., the event area). In order to achieve its goal, the proposed process performs a modified version of the Kruskal algorithm [33] for the minimum spanning tree problem. Briefly, given an indirect, connected and weighed graph, the Kruskal algorithm finds a minimum spanning tree, i.e., a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycle and with the minimum possible total edge weight.

In detail, during the candidate location computation phase the Kruskal algorithm is applied by considering the available geographic locations as vertices and the edges between the geographic locations as arcs. The trigonometric formula of Haversine [34] is used to assign a weight to each arc. In particular, it calculates the airline distance between two points on the Earth starting from their latitude and longitude. In order to remove outlier locations and obtain a specific target geographic area, the arcs with a weight greater than the average weight computed by considering all arcs, are

[3]https://github.com/michmech/lemmatization-lists/blob/master/lemmatization-it.txt
[4]https://github.com/michmech/lemmatization-lists/blob/master/lemmatization-en.txt
[5]https://www.urbandictionary.com/
[6]https://cloud.google.com/maps-platform/

deleted from the graph. The Kruskal algorithm initially sorts the arcs in ascending order according to their weight. Then, it analyzes them sequentially and inserts an arc into the final solution if it does not form cycles with previously selected arcs. The original Kruskal algorithm ends when the number of arcs included in the final solution is equal to the number of vertices minus one. However, during the target area computation phase, the executed Kruskal algorithm uses a different termination criterion. In detail, it stops the execution as soon as the final solution represents a tree with a number of vertices greater than a given percentage σ of the number of vertices, with σ a threshold chosen by the user. All vertices, i.e., the geographic locations, composing the final solution define the target geographic area where the event occurred. In order to identify this area, the Rectangle function in the Google Maps Javascript API[7] is used. This function allows one to define a rectangle on a map that includes all the coordinates given in input. An example of the geographic area defined by the function Rectangle is shown in Fig. 5. The red markers represent the vertices/locations inserted in the final solution computed by the Kruskal algorithm, whereas, the orange markers represent locations that are not inserted in the final solution and, as a consequence, they are considered outliers. The red rectangle represents the target geographic area computed by our approach by taking into account the red markers.

FIGURE 5. An example of a geographical area determined by red markers and the orange outlier markers.

IV. EXPERIMENTS AND RESULTS

This section is devoted to study the performance of the proposed process by means of a set of preliminary experiments.

A. DATASETS

The experiments involve three datasets collected by capturing Italian tweets in three time windows from April to June 2016. The capturing procedure has exploited the Phirehose library,[8] i.e., an open-source PHP implementation of the Twitter Stream API requirements. As other Twitter Stream APIs, Phirehose library enables to get a small amount, i.e. less than 1%, out of the total flow of tweets. The extraction of Italian tweets has been performed off-line in order to avoid to lose further tweets because of data processing. Table 1 shows the features of these datasets including day and time in which the datasets have been captured and the number of tweets that they contain. These datasets have been analyzed manually in order to obtain the ground truth data, i.e., the real events and the corresponding geo-locations. In detail, the collected tweets in each dataset have been analysed by two annotators. Each annotator gave the own list of events with the corresponding geo-locations for each set of tweets contained in the collected datasets. Annotators selected geo-locations according to the common geographic taxonomy such as city, country and so on since it is difficult for humans to produce a geospatial area outlined by coordinates. Starting from the two lists of events, the ground truth has been built by considering only the events identified by both annotators whose the identified geo-locations were the same. It is worth noting that annotators were aware of that an event is defined as a unique thing that happens at a specific time and place. Therefore, they left out the tweets that were not associated to a place such as those that describe user emotional states and those associated to more places such as, for instance, the women's day. As a consequence, these tweets were removed also by the analysis performed by the automatic proposed approach. Table 2 reports the events identified in each dataset together with their description and their geo-locations.

TABLE 1. Features related to the collected datasets of tweets.

Dataset	Day	Time	# Tweets
DB1	April 17, 2016	from 3:24pm to 6:39pm	1354
DB2	May 21, 2016	from 6:25pm to 10:16pm	1512
DB3	June 6, 2016	from 11:14am to 1:03pm	651

TABLE 2. Ground truth manually obtained starting from collected datasets.

Event	Dataset	Description	Geo-loaction
E1	DB1	Football match Palermo-Juventus	Palermo
E2	DB1	Football match Lazio-Empoli	Rome
E3	DB2	Final match of the Italian Cup	Rome
E4	DB3	Political election results in Rome	Rome
E5	DB3	Political Election Results in Naples	Naples

B. EXPERIMENT CONFIGURATION

All phases of the proposed process have been implemented in PHP[9] except for the tweet indexing phase and the clustering process that have been developed with the R[10] programming language. The proposed process has been executed on a laptop Macbook Pro 2010 with 4Gb of RAM and a processor Intel Core duo 2. As described in the previous section, our

[7] https://developers.google.com/maps/documentation/javascript/tutorial
[8] https://github.com/fennb/phirehose/wiki/Introduction
[9] http://www.php.net
[10] https://www.r-project.org

(a) Geographic Area for E1 (b) Geographic Area for E2 (c) Geographic Area for E3

(d) Geographic Area for E4 (e) Geographic Area for E5

FIGURE 6. In each figure, there is the geographic area computed by our approach in red and the effective geo-location of the event in blue.

approach requires only the specification of a value of the threshold σ. In these preliminary experiments, this value σ has been set to 20% experimentally by using toy datasets.

In order to evaluate our approach, we consider a *hit* when the computed geographic area contains the effective location of the event. In order to deepen the evaluation of our approach, similarly to other works [17], [24], in the case of a hit, we use an error metric to evaluate the quality of the computed geographic area. More in detail, we use the *maximum distance of error*, denoted as *MaxDist*, which represents the airline distance in kilometers between the point where the event is effectively located and the farthest point in the geographic area computed by our method. Formally, let $p(E)$ be the point where the event E is effectively located and $area(E) = \{p_1, p_2, \ldots, p_n\}$ the set of points of the perimeter of the geographic area computed by our approach for the event E, then the maximum distance of error for the event E, $MaxDist(E)$, is defined as follows:

$$MaxDist(E) = \max_{1 \leq j \leq n} (dist(p(E), p_j)) \text{ with } p_j \in area(E) \quad (9)$$

where *dist* is a function which computes the airline distance in kilometers between two points.

In order to further enrich the study of the performance of our approach, we also consider the metric Acc_d. It represents the accuracy of the proposed approach when the following consideration is made: an event E is correctly identified if $MaxDist(E)$ is less than a certain value d. Formally,

$$Acc_d = \frac{|\{e|MaxDist(e) <= d\}|}{N} \quad (10)$$

where N is the number of events. In our experimentation, we use three values for d, i.e., 250, 500 and 750, similarly to the work in [24].

C. RESULTS

Fig. 6 shows the geographic areas computed for each event contained in the collected data, whereas, Table 3 shows the performance of the proposed approach in terms of the metrics described above.

TABLE 3. Evaluation of the performance of our approach (*km* stands for kilometers).

Event	Dataset	Hit	MaxDist
E1	DB1	yes	542 km
E2	DB1	yes	681 km
E3	DB2	no	-
E4	DB3	yes	0 km
E5	DB3	yes	211 km

By analysing the Table 3, it is possible to state that the proposed process succeeds in the geo-localization of 4 out

of 5 events. In particular, the location of the event E3 is erroneously computed because the information about the real location is not present in the content of tweets. Moreover, it is not even present in the geo-located metadata attribute *place.name*.

To deepen the evaluation, Table 4 shows the accuracy Acc_d values. As shown in the Table, the proposed approach achieves an accuracy of about 80% by considering an error of 750 kilometers in the computation of the geographic area. Since 750 kilometers can be considered a reasonable value with respect to the earth surface, these preliminary experiments highlight the suitability of the proposed approach in geo-locating events discussed in Twitter.

TABLE 4. Performance in terms of accuracy in percentage.

$\mathbf{Acc_{d=250}}$	$\mathbf{Acc_{d=500}}$	$\mathbf{Acc_{d=750}}$
40%	40%	80%

TABLE 5. Evaluation of the performance of our approach with respect to literature approaches (***km*** stands for kilometers).

Event	Our Approach	ProFreq	LocFreq
E1	542 km	1,104 km (Italy)	1,104 km (Italy)
E2	681 km	748 km (Italy)	748 km (Italy)
E3	-	748 km (Italy)	748 km (Italy)
E4	0 km	0 km (Rome)	235 km (Lazio)
E5	211 km	940 km (Italy)	940 km (Italy)

D. COMPARISON WITH LITERATURE APPROACHES

In this last subsection, we compare the results obtained using the proposed approach with the results of two baseline methods existing in the literature in order to further show the feasibility of our proposal. In particular, we consider the following existing methods:

- *Maximum Profile Frequency* (ProFreq): this approach consists of finding the frequencies of locations specified in the profiles of the users that posted a tweet about an event. The location mostly referred to in users' profiles is assigned as the event location. The implementation of this method is inspired to the work of Giridhar *et al.* [35] and it is already used as baseline in [36];
- *Maximum Location Frequency* (LocFreq): this approach consists of finding the frequencies of locations specified by the users when they posted a tweet about an event, namely the frequencies of the values of the attribute *place.name*. The location value mostly specified is assigned as the event location. This method is a variant of the baseline approach already used in Sakaki *et al.* [37] where the mean of GPS coordinates are considered. In this work, we use the values in the attribute *place.name* being this attribute more typically set by users rather than GPS coordinates.

By running the methods ProFreq and LocFreq, the obtained location for the events is almost always the region area "Italy". Indeed, Italian users usually enter "Italy" for their own locations during the registration procedure or for the location in the *place.name* because they want to speed up the writing of the tweets or they do not want to give information about themselves. For this reason, with respect to our approach, the two baseline methods obtain a hit also for the event E3. On the other hand, the *maxDist* obtained by ProFreq and LocFreq are always worse or equal than those obtained by our approach. By speaking about the accuracy, as shown in Table 6, our approach is characterized by 80% of accuracy by considering 750 km for *maxDist*, whereas, ProFreq and LocFreq methods are characterized by 60% of accuracy.

TABLE 6. Comparison with literature approaches in terms of accuracy in percentage.

Method	$\mathbf{Acc_{d=250}}$	$\mathbf{Acc_{d=500}}$	$\mathbf{Acc_{d=750}}$
Our Approach	40%	40%	80%
ProFreq	20%	20%	60%
LocFreq	20%	20%	60%

V. CONCLUSION

This paper proposes a new approach to geo-localize events discussed in Twitter which relies on both textual and spatial/temporal use of content posted on Twitter without using some prior knowledge about the events to be located. As shown by a set of experiments, the proposed approach achieves an accuracy of 80% by considering a tolerable error in the computation of the geographic areas. The proposed approach outperforms two existing methods in terms of accuracy. In the future, we plan some improvements of the proposed approach with respect to the following aspects:

- *Removal of tweets starts from bots and spammers*: the approach could detect bots and spammers accounts and then eliminate tweets published by them with the purpose of reducing noise.
- *Control and correction of the text*: the approach could consider the frequency of incorrect spelling in the tweets.
- *Normalization of hashtags*: in the proposed approach hashtags are simply treated as common words. An improvement could perform an hashtag normalization in order to segment these hashtags into more words.
- *Scalability*: by analysing the steps of our procedure, the most computationally expensive phases are the clustering procedure and the candidate location extraction that includes a comparison with words contained in a set of dictionaries. The computational effort of these two steps could increase when the system will be put in production (i.e., by considering a larger number of tweets). To address this issue, emerging big-data frameworks such as Apache Hadoop[11] could be used.
- *Sentiment analysis*: a data source like the one provided by Twitter, lends itself well to the application of sentiment analysis, i.e., applying techniques to identify and evaluate subjective elements within a written text. It could be interesting to use these techniques to identify the users' emotions and opinions with respect to an event.

[11] https://www.ibm.com/analytics/hadoop

REFERENCES

[1] T. Cheng and T. Wicks, "Event detection using Twitter: A spatio-temporal approach," *PLoS ONE*, vol. 9, no. 6, Jun. 2014, Art. no. e97807.
[2] O. Ozdikis, H. Oğuztüzün, and P. Karagoz, "A survey on location estimation techniques for events detected in Twitter," *Knowl. Inf. Syst.*, vol. 52, no. 2, pp. 291–339, Aug. 2017.
[3] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study: Final report," in *Proc. DARPA Broadcast News Transcription Understand. Workshop*, Lansdowne, VA, USA, Feb. 1998, pp. 194–218.
[4] F. Atefeh and W. Khreich, "A survey of techniques for event detection in Twitter," *Comput. Intell.*, vol. 31, no. 1, pp. 132–164, Feb. 2015.
[5] B. Huang and K. M. Carley, "A large-scale empirical study of geotagging behavior on Twitter," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, F. Spezzano, W. Chen, and X. Xiao, Eds., Vancouver, BC, Canada, Aug. 2019, pp. 365–373, doi: 10.1145/3341161.3342870.
[6] A. Feizollah, S. Ainin, N. B. Anuar, N. A. B. Abdullah, and M. Hazim, "Halal products on Twitter: Data extraction and sentiment analysis using stack of deep learning algorithms," *IEEE Access*, vol. 7, pp. 83354–83362, 2019.
[7] D. Yu, D. Xu, D. Wang, and Z. Ni, "Hierarchical topic modeling of Twitter data for online analytical processing," *IEEE Access*, vol. 7, pp. 12373–12385, 2019.
[8] K. Sato, J. Wang, and Z. Cheng, "Credibility evaluation of Twitter-based event detection by a mixing analysis of heterogeneous data," *IEEE Access*, vol. 7, pp. 1095–1106, 2019.
[9] M. Xu, X. Zhang, and L. Guo, "Jointly detecting and extracting social events from Twitter using gated BiLSTM-CRF," *IEEE Access*, vol. 7, pp. 148462–148471, 2019.
[10] M. Garg and M. Kumar, "Review on event detection techniques in social multimedia," *Online Inf. Rev.*, vol. 40, no. 3, pp. 347–361, Jun. 2016.
[11] Y. Huang, Y. Li, and J. Shan, "Spatial-temporal event detection from geo-tagged tweets," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 4, p. 150, Apr. 2018, doi: 10.3390/ijgi7040150.
[12] A. Mourad, F. Scholer, W. Magdy, and M. Sanderson, "A practical guide for the effective evaluation of Twitter user geolocation," *ACM Trans. Social Comput.*, vol. 2, no. 3, pp. 1–23, Dec. 2019, doi: 10.1145/3352572.
[13] A. Rahimi, T. Cohn, and T. Baldwin, "Twitter user geolocation using a unified text and network prediction model," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.* Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 630–636. [Online]. Available: https://www.aclweb.org/anthology/P15-2104
[14] A. Rahimi, T. Cohn, and T. Baldwin, "Semi-supervised user geolocation via graph convolutional networks," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*. Melbourne, VIC, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2009–2019. [Online]. Available: https://www.aclweb.org/anthology/P18-1187
[15] S. Apreleva and A. Cantarero, "Predicting the location of users on Twitter from low density graphs," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2015, pp. 976–983.
[16] E. Rodrigues, R. Assunção, G. L. Pappa, R. Miranda, and W. Meira, "Uncovering the location of Twitter users," in *Proc. Brazilian Conf. Intell. Syst.*, Oct. 2013, pp. 237–241.
[17] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: A content-based approach to geo-locating Twitter users," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2010, pp. 759–768.
[18] O. Ajao, J. Hong, and W. Liu, "A survey of location inference techniques on Twitter," *J. Inf. Sci.*, vol. 41, no. 6, pp. 855–864, Dec. 2015.
[19] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: A survey," *ACM Comput. Surv.*, vol. 47, no. 4, p. 67, 2015.
[20] J. Ao, P. Zhang, and Y. Cao, "Estimating the locations of emergency events from Twitter streams," *Procedia Comput. Sci.*, vol. 31, pp. 731–739, Jan. 2014.
[21] P. Giridhar, T. Abdelzaher, J. George, and L. Kaplan, "Event localization and visualization in social networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2015, pp. 35–36.
[22] O. Ozdikis, H. Oguztuzun, and P. Karagoz, "Evidential location estimation for events detected in Twitter," in *Proc. 7th Workshop Geograph. Inf. Retr. (GIR)*, 2013, pp. 9–16.
[23] S. M. Paradesi, "Geotagging tweets using their content," in *Proc. 24th Int. Florida Artif. Intell. Res. Soc. Conf. (FLAIRS)*, 2011, pp. 355–356.
[24] S. Khanwalkar, M. Seldin, A. Srivastava, A. Kumar, and S. Colbath, "Content-based geo-location detection for placing tweets pertaining to trending news on map," in *Proc. 4th Int. Workshop Mining Ubiquitous Social Environ.*, 2013, p. 37.
[25] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975.
[26] F. Barbieri, F. Ronzano, and H. Saggion, "What does this emoji mean? A vector space skip-gram model for Twitter emojis," in *Proc. LREC*, 2016, pp. 3967–3972.
[27] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, Mar. 1980.
[28] S. Yu, D. Berry, and J. Bisbal, "Performance analysis and assessment of a tf-idf based archetype-SNOMED-CT binding algorithm," in *Proc. 24th Int. Symp. Comput.-Based Med. Syst. (CBMS)*, Jun. 2011, pp. 1–6.
[29] L. Kaufman and P. Rousseeuw, *Clustering by Means of Medoids*. Amsterdam, The Netherlands: North Holland, 1987.
[30] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.
[31] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "Twitinfo: Aggregating and visualizing microblogs for event exploration," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2011, pp. 227–236.
[32] S. H. Burton, K. W. Tanner, C. G. Giraud-Carrier, J. H. West, and M. D. Barnes, "'Right time, right place' health communication on Twitter: Value and accuracy of location information," *J. Med. Internet Res.*, vol. 14, no. 6, p. e156, 2012.
[33] J. B. Kruskal, Jr., "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. Amer. Math. Soc.*, vol. 7, no. 1, pp. 48–50, Feb. 1956.
[34] G. Van Brummelen, *Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry*. Princeton, NJ, USA: Princeton Univ. Press, 2012.
[35] P. Giridhar, T. Abdelzaher, J. George, and L. Kaplan, "On quality of event localization from social network feeds," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2015, pp. 75–80.
[36] O. Ozdikis, H. Oğuztüzün, and P. Karagoz, "Evidential estimation of event locations in microblogs using the Dempster–Shafer theory," *Inf. Process. Manage.*, vol. 52, no. 6, pp. 1227–1246, Nov. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306457316301900X
[37] T. Sakaki, M. Okazaki, and Y. Matsuo, "Tweet analysis for real-time event detection and earthquake reporting system development," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 919–931, Apr. 2013.

GIOVANNI ACAMPORA (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Salerno, Fisciano, Italy, in 2007.

From July 2011 to August 2012, he was a Hoofddocent Tenure Track of process intelligence with the School of Industrial Engineering, Information Systems, Eindhoven University of Technology, Eindhoven, The Netherlands. He was a Reader of computational intelligence from the School of Science and Technology, Nottingham Trent University, Nottingham, U.K., from September 2012 to June 2016. Since 2016, he has been an Associate Professor of artificial intelligence with the University of Naples Federico II. He is the Chair of IEEE-SA 1855WG, the working group that has published the first IEEE standard in the area of fuzzy logic. His main research interests include computational intelligence, fuzzy modeling, evolutionary computation, and ambient intelligence. Prof. Acampora is a member of the scientific board of the Interdepartmental Center for Advanced RObotics in Surgery (ICAROS). He was a recipient of two prestigious awards: the IEEE-SA Emerging Technology Award in 2016 and the 2019 Canada-Italy Innovation Award for Emerging Technologies. In 2017, he acted as a General Chair of IEEE International Conference on Fuzzy Systems, the top leading conference in the area of fuzzy logic. He serves as an Editor in Chief of Springer *Quantum Machine Intelligence*, an Associate Editor of Springer *Soft Computing*, and an editorial board member of Springer *Memetic Computing*, Elsevier *Heliyon*, Inderscience *International Journal of Autonomous and Adaptive Communication Systems*, and the IEEE TRANSACTIONS ON FUZZY SYSTEMS.

PAOLO ANASTASIO received the bachelor's degree in computer science and the master's degree from the University of Salerno, Fisciano, Italy, in 2016 and 2018, respectively.

He is currently an IAM and API Security Specialist with Spike Reply, Milan, Italy. His interests include big-data analysis, telecommunication network management, network protocols, and cybersecurity.

MICHELE RISI (Member, IEEE) received the Laurea degree in computer science and the Ph.D. degree in computer science from the University of Salerno, Italy, in 2001 and 2005, respectively.

Since 2019, he has been an Associate Professor with the Department of Computer Science, University of Salerno. He has published more than 100 articles on these topics in international journals, books, and conference and workshop proceedings. He has served as a program committee member of several international conferences. His research interests include reverse engineering (architecture and design pattern recovery), reengineering, human–computer interaction, empirical software engineering, big-data analysis, data-warehouse and data visualization, visual languages (visual programming environment, parsing techniques, and sketch understanding), mobile development and applications, and robot programming. He is a member of the review board and an editorial board of international journals.

GENOVEFFA TORTORA (Senior Member, IEEE) received the Laurea degree in computer science from the University of Salerno.

From 1978 to 1998, she was with the Department of Computer Science. Since 1990, she has also been a Full Professor of computer science. In 1998, she founded the Department of Mathematics and Computer Science and has been the Department Chair. From 2000 to 2008, she was the Dean of the Faculty of Mathematical, Physical and Natural Sciences, University of Salerno. Her research interests include software engineering, image processing and biometric systems, human–computer interaction, visual languages, databases, datawarehouses, and geographic information systems. From November 1999 to October 2000, she was a member of the board of director (Consiglio di Amministrazione) of the University of Salerno. Since November 2015, she has also been a member of the board of director (Consiglio di Amministrazione) of Fondazione Ravello. Since December 2015, she has also been a member of CNGR (National Committee of Research Guarantors) of the Italian Ministry of Education, University and Research. She is currently an Associate Editor of the *International Journal of Software Engineering and Knowledge Engineering* and the *Journal of Visual Languages and Computing*.

AUTILIA VITIELLO (Member, IEEE) received the M.S. degree *(cum laude)* in computer science from the University of Salerno, in July 2009, defending a thesis in Time Sensitive Fuzzy Agents: formal model and implementation, and the Ph.D. degree in computer science from the University of Salerno, in April 2013, defending a thesis titled Memetic Algorithms for Ontology Alignment.

Since 2018, she has been an Assistant Professor with the Department of Physics ''Ettore Pancini'', University of Naples Federico II. She is currently the Vice-Chair of the IEEE CIS Standards Committee and the Chair of the Task Force named Datasets for Computational Intelligence Applications. She is part of the IEEE Standard Association 1855 Working Group for Fuzzy Markup Language Standardization, where she also serves as Secretary. Her main research interests include computational intelligence, and in particular, fuzzy logic and evolutionary algorithms. Her recent interests include the integration between computational intelligence and computer vision to address bloodstain pattern analysis and the integration between evolutionary algorithms and machine learning techniques to tackle big data challenges.

Dr. Vitiello was a recipient of the Best Paper Award at the United Kingdom Workshop on Computational Intelligence, UKCI 2012, Edinburgh, U.K.