

Received July 24, 2020, accepted August 23, 2020, date of publication September 9, 2020, date of current version September 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022834

Detection of Barriers to Mobility in the Smart City Using Twitter

MARIO SÁNCHEZ-ÁVILA¹, MARCOS ANTONIO MOURIÑO-GARCÍA¹, JESÚS A. FISTEUS¹,
AND LUIS SÁNCHEZ-FERNÁNDEZ^{1,2}

¹Department of Telematic Engineering, Universidad Carlos III de Madrid, 28911 Leganés, Spain

²UC3M-Santander Big Data Institute, Universidad Carlos III de Madrid, 28911 Leganés, Spain

Corresponding author: Luis Sánchez-Fernández (luiss@it.uc3m.es)

This work was supported in part by the Analytics Using Sensor Data for FLATCity Project (Ministerio de Ciencia, innovación y Universidades/ERDF, EU) funded by the Spanish Agencia Estatal de Investigación (AEI), under Grant TIN2016-77158-C4-1-R, and in part by the European Regional Development Fund (ERDF).


ABSTRACT We present a system that analyzes data extracted from the microblogging site Twitter to detect the occurrence of events and obstacles that can affect pedestrian mobility, with a special focus on people with impaired mobility. First, the system extracts tweets that match certain predefined terms. Then, it obtains location information from them by using the location provided by Twitter when available, as well as searching the text of the tweet for locations. Finally, it applies natural language processing techniques to confirm that an actual event that affects mobility is reported and extract its properties (which urban element is affected and how). We also present some empirical results that validate the feasibility of our approach.

INDEX TERMS Mobility barriers, smartcity, social sensing, transport, twitter.

I. INTRODUCTION

Smart cities aim to use Information and Communication Technologies (ICTs) to improve the life of citizens in urban environments [1]. One aspect where technology can greatly ease the life of citizens is mobility.

With the use of GPS devices plus detailed maps, current technology is able to recommend routes to citizens. There exist, however, some aspects in which this technology could be improved. First, it should be possible to produce routes adapted to the needs of citizens that have some limitations with respect to their mobility capabilities (wheelchair riders, older people, pregnant women, etc.). Second, maps should be kept as up to date as possible. Although mobile cameras and LIDAR devices can be used to capture data, this process is costly. Third, these systems should be able to detect events that affect mobility, such as potholes, pavement in poor condition, slippery surfaces, broken trees, etc., and take them into account in their route recommendations. Early detection of these events would help not only for route recommendation but also to help authorities fix them sooner. Early detection of these events with cameras and LIDAR devices would be costly because it would require a continuous monitoring of the streets with such devices.

The associate editor coordinating the review of this manuscript and approving it for publication was Imran Sarwar Bajwa .

As will be explained in Section VI, the Twitter social network has been successfully used in the past as a kind of *social sensor* to detect, in a timely manner, issues in related areas such as public transportation systems [2] and road traffic [3]–[7].

Similarly, in this work we also use Twitter as a social sensor. In particular, we present a system that monitors that social network and analyzes its tweets in order to detect events and obstacles that can affect pedestrian mobility, with special focus on people with impaired mobility. The system gets activated whenever a new tweet matching certain predefined terms is detected. Then, it obtains location information from the tweet by using the location provided by the metadata of the tweet, when available, as well as by searching the text of the tweet for mentions to some specific location. Finally, it applies natural language processing (NLP) techniques to confirm that an event that affects mobility is actually reported in the tweet and extract its properties (which urban element is affected and how). To our knowledge, a system like the one proposed in this article for the detection of mobility barriers in smart cities has not been proposed before. See the Related Work section for a more detailed discussion.

The system has been developed to process tweets written in Spanish with information regarding mobility events located in Spain, although its architecture could be applied to

other languages and locations with a reasonable adaptation effort.

The rest of the paper is organized as follows. Section II describes the architecture of the proposed application. The location detection algorithm is described in Section III and the NLP tool that estimates the validity of the tweet is presented in Section IV. Section V will be devoted to evaluate this NLP tool as well as the whole system presented with an experiment in which we captured 30,000 tweets. Finally, Sections VI and VII are devoted, respectively, to present some work related to ours and to draw some conclusions.

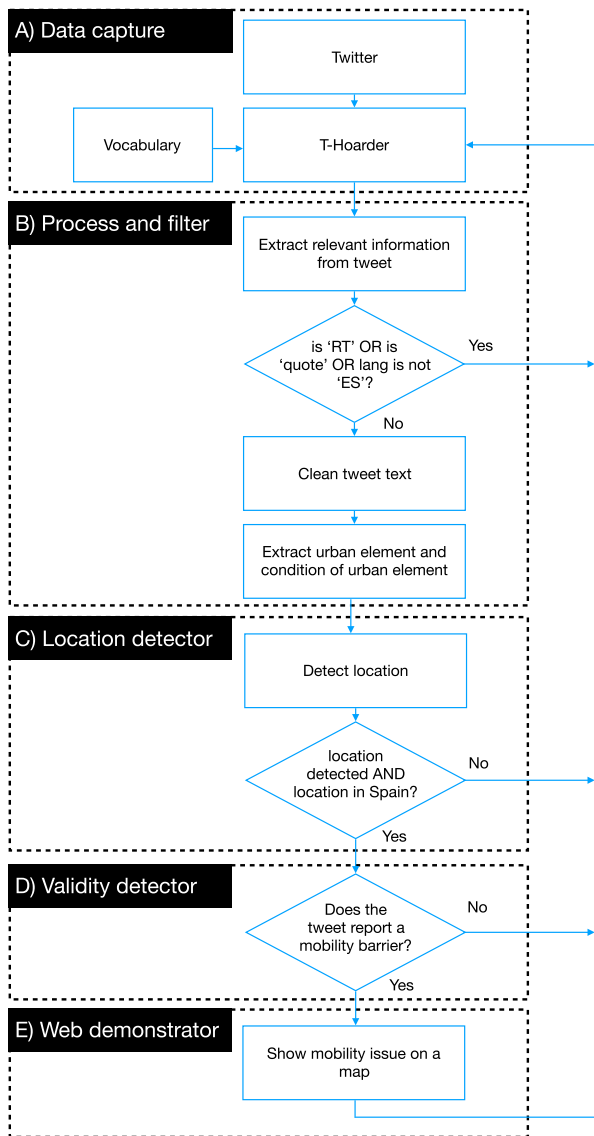


FIGURE 1. Architecture and flowchart of the system proposed to detect the occurrence of events that can affect mobility.

II. ARCHITECTURE

Figure 1 shows the architecture of the proposed system, which is composed of the following blocks: A) Data capture,

B) Data processing and filtering of non-relevant tweets, C) Location detection, D) Validity detection, and E) Web demonstrator.

In short, the process of capturing and extracting relevant information from a tweet consists of the following steps:

- 1) A tweet is captured if it matches some terms in a preset vocabulary of urban elements and conditions.
- 2) If the tweet is written in Spanish and is not a retweet or a quote, the process and filter block cleans its text from undesired symbols and extracts the urban element and condition it mentions. Otherwise, the tweet is discarded.
- 3) The location detector module obtains the location of the tweet. If the module is not able to obtain the location or its location is outside Spain, the tweet is discarded.
- 4) The validity detector performs a final check about whether the tweet actually reports a mobility barrier or mobility issue. If not, the tweet is discarded.
- 5) The mobility issue is reported and shown on the map of the web demonstrator.

The rest of this section describes the above mentioned blocks in more detail.

A. DATA CAPTURE

In order to extract the data (i.e. tweets) from the microblogging site Twitter, we used the T-Hoarder tool [8]. T-Hoarder allows to capture tweets about a specific topic by matching certain predefined terms included in a vocabulary that has to be provided to the tool.

The vocabulary has to be selected according to the topics about which tweets need to be captured. In the case of our system, we are interested in capturing tweets about issues that can affect mobility. To that end, we selected two types of terms to generate the vocabulary:

- **Urban elements:** “acera”, “paso de cebra”, “loseta”, “suelo”, etc. (English translation: *pavement, pedestrian crossing, tile, ground*).
- **Condition of the urban elements:** “mal estado”, “estrecho”, “resbaladizo”, “agujero”, etc. (English translation: *poor condition, narrow, slippery, hole*).

The vocabulary provided to T-Hoarder will be composed by all the possible combinations of urban elements and element condition. In particular, we composed a vocabulary consisting of 512 combinations. Some examples of combinations were: “pavimento en mal estado”, “paso de cebra en mal estado”, “pavimento estrecho”, “paso de cebra estrecho”. (English translation: *pavement in poor condition, pedestrian crossing in poor condition, narrow pavement, narrow pedestrian crossing*).

T-Hoarder keeps a live connection to the Twitter streaming API and produces as output, as soon as they appear in the stream, the tweets that match that vocabulary. For each tweet, it provides a total of 28 fields of information. Among them, the most significant ones for the purpose of our system are: text of the tweet; publication date and time; unique identifier of the tweet; language identifier; whether the tweet is a

retweet (RT), a quote or a normal tweet; location from its author's profile; georeferencing information; and geolocation information.

T-Hoarder provides three types of location information for tweets, depending on how they were created:

- Non-geolocated tweets, which do not contain any location information.
- Geolocated tweets, which contain the precise coordinates of the place from where the tweet was published.
- Georeferenced tweets, which refer to a place, such as a park, a neighborhood, a city, etc. without precise coordinates.

B. DATA PROCESSING AND FILTERING OF NON-RELEVANT TWEETS

This block processes and performs a first filtering of the tweets provided by T-Hoarder. More specifically:

- It performs a first filtering of the tweets captured by the T-Hoarder tool based on the value of some of their fields. More specifically, it discards tweets that are retweets or quotes, and tweets that are not written in Spanish.
- It removes from the text of the tweets symbols such as parentheses, brackets, at and hashtag signs, exclamation and question marks, emoticons and URLs.
- It identifies the urban element mentioned in the text of the tweet and its condition.

C. LOCATION DETECTOR

The location detector block is responsible for obtaining the location of the tweets. In addition, it discards tweets whose location could not be obtained or whose location is outside Spain. The way this block works is described in more detail in Section III.

D. VALIDITY DETECTOR

The validity detector block is in charge of checking whether the captured tweets are actually related to a mobility event. This block is described in more detail in Section IV.

E. WEB DEMONSTRATOR

The web demonstrator displays on top of a map all the incidents obtained from the analysis of the tweets. The Python Flask framework has been used to create the web application.

Figure 2 shows all the mobility incidents detected in Spain by the system during a specific time period. When clicking on a marker, a box appears and shows information about the urban element and its condition. As an example, Figure 3a) shows a mobility issue on Calle Daniel Segovia in Madrid that refers to a sidewalk with a hole. Double clicking on the marker opens a new window that redirects the user to the original tweet from which the issue was extracted, as can be seen in Figure 3b).

III. LOCATION EXTRACTION

Being able to determine the location of the mobility barrier a tweet reports is a crucial part of our system. As was already

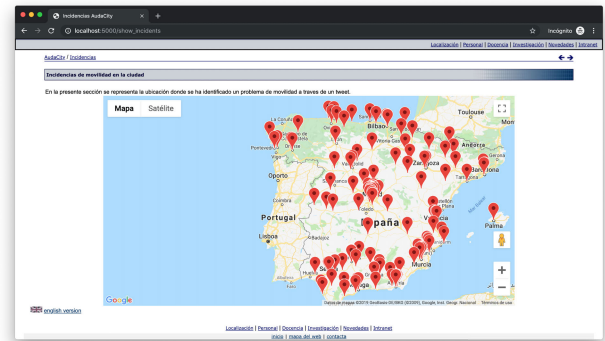


FIGURE 2. Web application.



FIGURE 3. Mobility issue example.

mentioned in Section II-A, T-Hoarder can provide, for each tweet, the following location information: only geolocation information (GPS coordinates), only georeferencing information, geolocation and georeferencing information or, in a majority of the tweets, no location information.

The Location Extraction block of the system is in charge of obtaining the location of the mobility barrier (GPS coordinates) from the tweet itself in any of those three cases. The operation of this module is as follows:

- If the tweet is geolocated, it gets its GPS coordinates from that field.
- If the tweet is georeferenced, it gets the reference from that field and uses the HERE API Geocoder [9] to obtain GPS coordinates.
- In all cases, it applies NLP techniques to the text of the tweet in order to obtain location information (e.g. a street address), and uses the HERE API Geocoder to get GPS coordinates from it. This is further explained in section III-A.

The procedure above could yield more than one location from the same tweet (e.g. one location from the geolocation field and another one from the text of the tweet), section III-A1 explains how the system chooses a location when this happens.

A. OBTAINING LOCATION INFORMATION FROM THE TEXT OF THE TWEET

Our experiments showed that approximately 98.5% of the tweets captured in Spain that reported mobility issues were neither geolocated nor georeferenced. However, their authors

usually reported the location of the mobility barrier by including in the text of the tweet the street address or, at least, street name where the mobility issue is happening.

In order to leverage the aforementioned information, we implemented an algorithm that makes use of Natural Language Processing techniques and regular expressions to identify street addresses in the text of tweets.

We use the SpaCy Python library for Named Entity Recognition (NER) and Part-Of-Speech tagging. In an initial screening performed with a reduced set of tweets written in Spanish, this library showed a good performance in the detection of location named entities.

The NER module of the SpaCy library allows the identification of the following entities:

- PER: Named persons or family.
- LOC: Name of political or geographical locations (cities, provinces, countries, regions, etc.)
- ORG: Named corporations, governmental units, or other organizational entities.
- MISC: Miscellaneous entities (e.g. events, nationalities, products or works of art).

1) DETECTING THE STREET FROM THE TEXT OF THE TWEET

First, through the use of the Python SpaCy library, the system identifies entities of type LOC. The SpaCy library can provide more than one entity of that type from a piece of text.

Besides, we built a database with geographical information of Spain obtained from the Spanish National Statistics Institute (INE) [10], which included city and town names, street names belonging to each one, etc. The use of this database is of great importance for this project, since very fast searches can be carried out on streets, municipalities or provinces, allowing the results provided by the SpaCy module to be compared with the information from the database.

The street map of the electoral roll provided by the National Statistics Institute (INE) contains information of the whole street map of Spain, such as the name of autonomous communities, provinces, municipalities, street names, type of roads, postal codes, and districts. This information is processed and entered into a database in order to make queries in a quick and efficient way. The street map of the electoral roll is composed of four ASCII files: road file, pseudo-road file, road section file and population unit. More specifically, the file that has been used to extract all the streets of Spain with its corresponding municipality, province and postal code is the file of road sections, where each line of the file corresponds to a street and all its associated information.

In order to filter the results provided by SpaCy, we applied regular expressions to the different entities detected by the SpaCy library in order to check whether the entities detected by SpaCy referred to a street or not. Such regular expressions were based on the appearance of street type labels in the entity. Some examples of street type labels are: *calle*, *avenida*, *carretera*, *bulevar*, *cruce*, etc. (English translation: street, avenue, road, boulevard, junction).

There were certain occasions in which the previous procedure did not obtain information about a street. In those cases, we performed an additional full text search in the full text of the tweet by using the same regular expressions mentioned above in order to detect location information.

2) DETECTING THE CITY

If a street was identified by any of the two previous techniques, we queried the INE database in order to obtain the list of cities and villages where a street with such name exists.

If more than one city name was obtained, we proceeded to search the name of each of those cities in the text of the tweet in order to choose one. If none of them appeared, a new text search was done for the names of the provinces to which each city belonged.

Nevertheless, with the mechanisms used so far there were certain tweets from which no street address could be obtained. In order to solve this problem, we created a second algorithm. This algorithm analyzes first whether, after running the previous algorithm, the street or city name were obtained. Here we can distinguish two possible cases:

- It obtained the street name but not the city name.
- It obtained neither the street nor the city name.

In the first case, the location information from the Twitter user's profile of the author of the tweet was used. The city name was obtained by applying the geocoder to the postal address declared in the user's profile.

In the second case, in which neither street nor city name were found, the city name was obtained from the user's profile as explained in the paragraph above. Then, the system tried to identify the street name by querying the INE database for that city with the text of every entity of type LOC that SpaCy returned.

B. SELECTING THE MOST LIKELY LOCATION

After performing the previous steps, each tweet could potentially have up to three sources of location information: the geolocation field, the georeference field and the location obtained by NLP techniques. When more than one location was obtained for the same tweet, a decision was needed in order to estimate which location was the most accurate one. We developed another algorithm to that end:

- If there is only one source of location, that source is selected.
- If there are two sources of location, we selected the one most likely to be accurate according to the source of the location: (1) geolocation, (2) NLP techniques and (3) georeference. On the one hand, we expect geolocation information to be the most accurate since Twitter directly provides the GPS coordinates of the device from which the tweet has been published. We assume tweets will be often created close to the location of the reported mobility issue. On the other hand, georeference information is probably the least accurate one because it

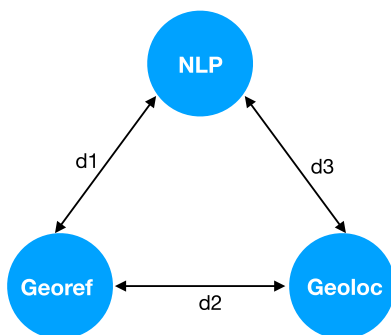


FIGURE 4. Distance between sources of tweet location information.

corresponds to an area on the map and not some specific GPS coordinate point.

- If there are three sources of location information, we first compute the distance between each pair of geographical points by using the Haversine equation [11]. Then, from the two points that were closest together, we selected the one most likely to be accurate as explained above (see Figure 4).

IV. VALIDITY DETECTOR

As previously mentioned, the T-Hoarder system collects tweets that report the occurrence of events that can affect mobility by matching certain predefined terms that represent urban elements (e.g. pavement, tile, zebra crossing, etc.) and the condition of these urban elements (e.g poor condition, slippery, narrow, etc.). But in some cases, the appearance of these terms in the text of the tweet does not necessarily imply that the tweet is reporting a mobility barrier. The following example shows two tweets that match terms that represent urban elements and the the condition of these elements. The first tweet reports a mobility barrier but the second does not.

Tweet #1: *El paso de cebra de la calle Sabatini se encuentra en mal estado.* (English translation: *The pedestrian crossing of Sabatini street is in poor condition*). In this case, the tweet matches the urban element “paso de cebra” (zebra crossing) and the condition of the urban element “mal estado” (poor condition), and indeed it reports a mobility barrier.

Tweet #2: *El ayuntamiento declaró interés departamental la planta de UPM2 que contaminará nuestro suelo y agua, y dejará un agujero económico.* (English translation: *The city council declared departmental interest the UPM2 plant that will pollute our ground and water, and leave an economic hole*). In this case, although the tweet matches a term that identifies an urban element (“suelo”, ground) and a term that represents the condition of an urban element (“agujero”, hole), the tweet does not report any mobility barrier.

In order to exclude such tweets that do not represent a mobility barrier, we developed a machine learning classifier that was trained with examples of what a relevant tweet is (i.e., a tweet that reports a mobility barrier) and examples of

what a non-relevant tweet is (a tweet that does not report any mobility barrier).

A. DOCUMENT (TWEET) REPRESENTATION

Documents (tweets in this case) have to be represented in such a way that the classifier can understand and relate them. One of the most used representations in text classification tasks is the vector space model (VSM) [12], according to which each document in a collection is represented as a vector, and each dimension of the vector represents a feature. Thus, following the vector space model, a document is represented as a vector $\vec{d} = (w_{f_1}, w_{f_2}, \dots, w_{f_n})$ where w_{f_i} is the weight of feature f_i in document \vec{d} .

In order to select features, we used natural language processing techniques on the text of the tweets and, more specifically, we employed a hybrid approach by combining manually selected features and automatically extracted features from the text of the tweets.

On the one hand, the features we selected manually are those presented in Table 1, where the name and description of each one is shown.

On the other hand, the features that were automatically extracted from tweets were obtained by using the Wikipedia Miner semantic annotator [13]. Wikipedia Miner is a general purpose semantic annotator based on natural language processing, machine learning techniques, and the use of Wikipedia as background knowledge. This approach has been successfully applied in previous studies for the classification of, among others, biomedical documents [14], documents of legal nature [15], and news [16]. The main characteristics of Wikipedia Miner are: 1) It identifies concepts that appear in documents, thus avoiding the generation of irrelevant features; 2) it performs word sense disambiguation, thus tackling synonymy and polysemy problems; 3) it links the extracted concepts from documents to Wikipedia entries; and 4) it assigns a weight to each extracted concept according to its relevance in the text.

Figure 5 shows an example of the concepts extracted from a tweet by using Wikipedia Miner. These concepts will be used as features to enrich the representation of tweets, by assigning each feature a weight consisting in the relevance (weight) that Wikipedia Miner assigned to each one.

As a result, the hybrid representation of a tweet is the following vector:

$$\vec{d} = (w_{mf_1}, w_{mf_2}, \dots, w_{mf_n}, w_{af_1}, w_{af_2}, \dots, w_{af_m}) \quad (1)$$

where w_{mf_i} is the weight of a manually selected feature m_{f_i} , and w_{af_i} is the weight of a feature a_{f_i} that was automatically extracted by Wikipedia Miner.

V. EVALUATION OF THE SYSTEM

This section describes the evaluation of the system. First, the validity detector (the machine learning based block intended to automatically discard non-relevant tweets) is

TABLE 1. Manually selected features and their description.

#	Feature	Description
1	nwords	Number of tweet words
2	d_elem_cond	Distance between the urban element and the condition of the element
3	n_verb_total_bt_elem_cond	Number of verbs between the urban element and the condition of the element
4	n_verb_no_aux_bt_elem_cond	Number of non auxiliary verbs between the urban element and the condition of the element
5	n_verb_aux_bt_elem_cond	Number of auxiliary verbs between the urban element and the condition of the element
6	n_adj_bt_elem_cond	Number of adjectives between the urban element and the condition of the element
7	n_verb_total	Number of verbs in the tweet
8	n_verb_no_aux	Number of non auxiliary verbs in the tweet
9	n_verb_aux	Number of auxiliary verbs in the tweet
10	n_adj	Number of adjectives in the tweet
11	n_appearances_elem	Number of appearances of the urban element in the tweet
12	n_appearances_cond	Number of appearances of the condition of the urban element in the tweet



FIGURE 5. Wikipedia Miner concept extraction process.

evaluated in isolation. Then, the full system is evaluated as a whole.

A. EVALUATION OF THE VALIDITY DETECTOR

This section presents the corpus used and the experiments conducted to verify the performance of the machine classifier of the validity detector, as well as the results obtained and their analysis.

1) CORPUS

The corpus used to evaluate the performance of the classifier was composed of 560 tweets that were manually annotated by humans as relevant or non-relevant, depending on whether the tweet reported a mobility barrier or not. Particularly, the corpus is composed of 234 tweets annotated as relevant and 326 tweets annotated as non-relevant.

2) EXPERIMENTAL SETTINGS

The experiments consisted in training a machine learning classifier using the tweets from the corpus, in order to later automatically classify unlabeled tweets to decide

whether they represent a mobility barrier or not. To that end, we selected four of the most used, relevant, and best performing machine learning algorithms in automatic text classification tasks: Support Vector Machines (SVM), Random Forest (RF), Logistic Regression (LR), and Classification and Regression Trees (CART) [17], [18]. Of course, any other text classification algorithm like those based on neural networks (LSTM [19], BERT [20], etc.) could also have been used. The evaluation presented in this section shows that the text classification techniques considered in this work provide enough accuracy to validate the feasibility of our approach.

The implementation of those algorithms in Scikit-learn [21] version 0.21.2, a machine learning library for Python, was used. In particular, the following classes were selected: sklearn.svm.LinearSVC(), sklearn.ensemble.RandomForestClassifier(), sklearn.tree.DecisionTreeClassifier(), and sklearn.linear_model.LogisticRegression(). In all the experiments we used the default settings and parameters provided by the Scikit-learn library.

Besides, we combined the four selected algorithms with the application or not of standardization (normalization) techniques to the weights of features, and enriching or not the representation of tweets with the features extracted from Wikipedia using Wikipedia Miner, thus resulting in 16 different experiments.

In order to determine the performance of the classifier we selected the K-folds cross-validation strategy [22]. In particular we selected a 10-folds cross-validation strategy, so that in each iteration we used 504 documents for training the classification algorithm and 56 documents for testing. To avoid biases, the entire corpus was randomly shuffled before applying K-folds.

3) CLASSIFIER RESULTS AND ANALYSIS

The results of the experiments are presented in terms of Accuracy, Precision, Recall and F1-score, the harmonic

mean of Precision and Recall [23], [24]. Table 2 shows the averaged results of the 16 experiments performed after the 10 iterations. The results clearly show the following:

- The Random Forests algorithm is the one that shows the best results, since it outperforms the other algorithms regardless of the combination of normalization and enrichment options selected.
- The Logistic Regression algorithm is the one that shows the worst results, since the other algorithms outperform it for all combinations of normalization and enrichment options except for the combination (Normalization, Non-enrichment), where it shows results similar to Support Vector Machines.
- Although the Random Forest algorithm is the one that offers the best performance metrics, it is also the one that needs more time to be trained and to classify the tweets, as shown in Table 2.
- It can also be clearly seen that the performance metrics are higher when enriching the tweets representations with the features extracted from the text of the tweets by using Wikipedia Miner, achieving F1-score improvements up to 13% for the Random Forest algorithm and up to 22.23% for the CART algorithm. This is a clear evidence that the knowledge contained in Wikipedia provides very relevant information to the classifier, thus improving its performance, which is in line with what was stated in previous studies [14]–[16].
- Finally, after the analysis of the results presented, we concluded that the best option for this particular case is the CART algorithm, since it shows performance values similar to Random Forests with significantly lower training and classification times. Whereas the CART algorithm takes around two seconds for training and 2.46 milliseconds on average for classifying each tweet, Random Forest takes around 26 seconds for training the algorithm and 875 milliseconds on average for classifying each tweet. The averaged classification times per tweet are obtained by dividing the classification time presented in Table 2 by the number of elements for testing. As previously stated, we selected a 10-folds cross-validation strategy, so that in each iteration we used 504 documents for training the classification algorithm and 56 documents for testing.

B. EVALUATION OF THE SYSTEM AS A WHOLE

The main goal of the whole system is to identify and locate those tweets that report a mobility issue in Spain. This section presents its evaluation and validation.

To that end, we studied a set of tweets captured from May 7 to June 13, 2019. During that period, the Data capture module (block A of Figure 1) captured 30,006 tweets.

Following the flowchart depicted in Figure 1, the captured tweets are processed by block B, which is responsible for discarding retweets, quotes and tweets that are not written in Spanish. After this first processing, 1,717 tweets remain in the system (5.7%), representing 1,811 mobility issues.

This is because a single tweet can report more than one mobility issue.

Then, the remaining 1,811 issues were processed by the Location detector (block C), responsible for detecting the location of the issues. After this processing, 328 issues remained in the system: those for which it was possible to obtain their location and the location was in Spain.

Finally, the validity detector (block D) automatically discarded those tweets that did not represent a mobility issue. Among the aforementioned 328 issues, we randomly inspected 237 of them, where 148 were manually annotated as relevant (62.4%) and 89 as non-relevant (37.6%). For those 237 selected issues, the Validity detector was able to correctly decide if the tweet was a relevant tweet or not (that is to say, if the tweet reported a mobility barrier or not) in the 73.42%, 83.54%, 99.58% and 99.58% of cases by using the LR, SVM, CART, and RF classifiers respectively. These results are consistent with the values presented in Table 2, where the RF and CART algorithms offer the best performance, followed by SVM and LR.

VI. RELATED WORK

In this section we consider two types of related work. First, we study previous work related to the use of Twitter as a social sensor. Second, we review previous techniques that have been proposed to detect mobility barriers and events.

Social networks have a high volume of information that can be used for many applications. In order to use this data, there are different projects that allow real-time events to be detected from data captured from social networks.

Currently, Twitter is one of the most popular social networks. In fact, 500 million daily tweets about different topics are published, allowing studies in different social fields such as politics, social movements, natural disasters, influence and user behavior. This data can be captured and analyzed to obtain a specific objective. In fact, this information has been used by different research projects where events have been detected from Twitter data.

Sakaki *et al.* developed an earthquake notification system in Japan which, due to the large number of Twitter users that exist in the country, is able to detect 96% of the earthquakes detected by the Japan Meteorological Agency. The system is capable of notifying registered users on its platform faster than the ads of the Meteorology Agency itself [25].

Tweettronics is an application that analyzes tweets related to products and brands with marketing objectives, identifying the most influential users and the most important topics for users. In this way, companies can improve customer service and make more interesting offers, increasing the acquisition of potential customers [26].

Web2express Digest is a website that makes use of natural language processing and sentiment analysis to find what are the most interesting topics and trends of the moment, by processing data obtained from social media like Twitter, Facebook and LinkedIn and other web content. Companies

TABLE 2. Averaged accuracy, Precision, Recall, F1-score and training and classification times for the 16 experiments performed.

Algorithm	Normalization	Wikitopics	Accuracy	Precision	Recall	F1-score	Training time (ms)	Classification time (ms)
SVM	No	No	0.593	0.647	0.575	0.512	71	7
	Yes	No	0.675	0.664	0.657	0.657	45	8
	No	Yes	0.680	0.740	0.675	0.644	2089	124
	Yes	Yes	0.746	0.740	0.730	0.730	2038	123
CART	No	No	0.643	0.632	0.630	0.628	41	8
	Yes	No	0.686	0.682	0.685	0.675	42	9
	No	Yes	0.782	0.772	0.767	0.768	2243	134
	Yes	Yes	0.791	0.785	0.780	0.781	2077	123
LR	No	No	0.582	0.291	0.500	0.368	41	7
	Yes	No	0.670	0.668	0.648	0.645	48	9
	No	Yes	0.582	0.291	0.500	0.366	2066	125
	Yes	Yes	0.718	0.715	0.696	0.696	2046	123
RF	No	No	0.743	0.737	0.729	0.727	10611	49590
	Yes	No	0.720	0.714	0.709	0.707	10867	50737
	No	Yes	0.807	0.824	0.788	0.792	26107	49804
	Yes	Yes	0.814	0.828	0.794	0.799	25818	49198

can use this website for the effective control of brands and products [27].

There are other projects that have analyzed tweets for studying natural disasters such as the Oklahoma fire and the Red River floods in North America, allowing to identify the stages of the disaster [28].

In the context of smart cities, there have also been multiple proposals that gather information from Twitter and other social networks. Metro Averías is a project that analyzes complaints related to Madrid Metro, the underground transportation system of Madrid, with the objective of detecting breakdowns and problems in public transport in real time. In addition, the system disseminates its results graphically at the @metroaverias account to alert other users of Twitter of possible breakdowns [2].

Demirbas *et al.* propose the use of Twitter as a mechanism for crowdsensing dissemination, that is, to publish information extracted from sensors carried by Twitter users (typically mobile phones) [29].

Silva *et al.* use Twitter to collect Waze alerts about traffic incidents (i.e. a traffic jam) reported by Waze users [3].

Pan *et al.* first analyze human mobility data (i.e. GPS data) to detect traffic anomalies. Then, they extract tweets related to a given traffic anomaly. They first use temporal and spatial information to filter tweets of interest. Then, they compare the obtained tweets with historical tweets in the same location to select tweets that follow a different pattern and thus may be related to the traffic anomaly. With that they produce a sort of natural language explanation of the detected traffic anomaly [4].

The SMARTY project is presented in [5]. In this project data extracted from social networks (including Twitter) is used to detect events like accidents, demonstrations,

concerts, etc. that can affect traffic and parking requirements. The work of Kumar *et al.* pursue a similar goal: detecting traffic incidents (accidents, traffic jams, etc.) [6].

Finally, the work of Wanichayapong *et al.* is the most similar to ours that we have found. They use Twitter to find traffic events like accidents and traffic activity, but also obstruction hazards and road conditions. Similarly to us, they use a dictionary with Places (similar to our urban elements) and Verbs (similar to our Conditions of urban elements) to extract tweets related to traffic issues. In their case, they extract tweets written in Thai. A main difference between our work and the work of Wanichayapong *et al.* is in the procedure to detect whether a tweet describes a traffic issue. They used a simple filtering process: first, the tweet is required to contain both a Place and a Verb (this is similar to our approach), and then they discard tweets that contain Ban words (word in a set of vulgarity/profanity or a word with an interrogative meaning). They evaluated their approach with a dataset of 1,249 manually annotated tweets (497 related to traffic issues and 752 not related). Their filtering procedure classifies correctly 91.75% of the analyzed tweets [7]. This compares with our result of 99.58% using an RF classifier. Of course, it is not surprising that the use of machine learning techniques improves accuracy. Another difference between our work and the work of Wanichayapong *et al.* is that we focus our work on cities, and therefore we try to identify the city and street where the mobility incident occurs, while they focus their work on mobility incidents that happen on roads.

There exist also many more previous work to detect mobility barriers and events using physical sensors. For instance, this problem is of interest in robotics, where it is typically addressed with the help of a number of sensors [30]. Focusing on the field of smart cities, several works make

use of sensors to detect different issues related to mobility in smart cities. For instance, in [31] LIDAR data is used to detect zebra crossings.

Several works can be found related to the use of crowdsensing for detection of mobility related events in smart cities. In [32] a study of the use of data taken from the smartphones of the drivers to detect traffic jams is performed. Prandi *et al.* [33] propose a method to detect mobility barriers based on authoritative reports (maps or other data sources provided by official organizations), user reports (reports written by a citizen on a mobility barrier) and sensor reports (data taken from sensors, typically located in smartphones). In their model the reports written by citizens are entered in an on-purpose application. Cardonha *et al.* [34] propose a platform to detect mobility barriers based on the collection of images taken from mobile phones of the citizens and also by capturing other types of information (acceleration, orientation, geolocation). Another difference between this work and ours is that this work seems to be focused in incidents occurring in roads and highways, while our work is focused on incidents in cities, and thus we have to deal with city and street names.

VII. CONCLUSION AND FUTURE WORK

We believe that the experiments described in this article show that our system can be useful to detect mobility barriers and mobility issues in the smart city by extracting information from Twitter. Examples of applications that can exploit the data that our system extracts are GPS navigators that propose routes adapted to the conditions of each particular user (pregnant women, older people, wheelchair riders, asthmatic people, etc.) and as a mechanism to report mobility issues to public authorities.

A system like the one proposed in this article can be applied to the detection of other types of events (for instance, traffic jams) in smart cities, although specific adaptations (search terms, classification, etc.) and evaluation should be done for each particular application.

We have used three types of location information extracted from the tweets that we captured. While we expect geolocated tweets to usually (but not always) be very precise about the location of the mobility issue, the other two mechanisms are less reliable. In the case of locations extracted from the text of the tweet, we have found that they rarely provide the exact location of the mobility issue (for instance, street number). A similar problem happens with georeferenced tweets. Such tweets point to an area in the map and not a precise location. Still, we believe that this information is useful, and can be for instance exploited in combination with physical sensors like LIDAR devices or the sensors located in a mobile phone, because the information extracted from Twitter can be used to restrict the scanning area, thus saving money and time. In fact, we plan, as a continuation to this work, to explore the possibility of combining social (Twitter) and physical sensors for the detection of mobility issues.

Although our system has been developed to extract information from tweets written in Spanish and about mobility issues located in Spain, we believe that it could be adapted with a reasonable amount of effort to other locations and languages. The components that are language and/or location dependent are the named entity extractor and the database with street and city names. Finally, the validity detector should be trained for each specific language and a POS tagger for the given language would also be needed. However, it should be noted that we could take advantage of the multilingual nature of Wikipedia, and thus the use of Wikipedia Miner would be language independent to a large extent.

REFERENCES

- [1] R. G. Hollands, "Will the real smart city please stand up?: Intelligent, progressive or entrepreneurial?" *City*, vol. 12, no. 3, pp. 303–320, Dec. 2008.
- [2] M. Congosto, D. Fuentes-Lorenzo, and L. Sanchez, "Microbloggers as sensors for public transport breakdowns," *IEEE Internet Comput.*, vol. 19, no. 6, pp. 18–25, Nov. 2015.
- [3] T. H. Silva, P. O. V. De Melo, A. C. Viana, J. M. Almeida, J. Salles, and A. A. Loureiro, "Traffic condition is more than colored lines on a map: Characterization of Waze alerts," in *Proc. Int. Conf. Social Informat. Kyoto, Japan: Springer*, 2013, pp. 309–318.
- [4] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi, "Crowd sensing of traffic anomalies based on human mobility and social media," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (SIGSPATIAL)*, 2013, pp. 344–353.
- [5] G. Anastasi, M. Antonelli, A. Bechini, S. Brienza, E. D'Andrea, D. De Guglielmo, P. Ducange, B. Lazznerini, F. Marcelloni, and A. Segatori, "Urban and social sensing for sustainable mobility in smart cities," in *Proc. Sustain. Internet ICT Sustainability (SustainIT)*, Oct. 2013, pp. 1–4.
- [6] A. Kumar, M. Jiang, and Y. Fang, "Where not to go?: Detecting road hazards using Twitter," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2014, pp. 1223–1226.
- [7] N. Wanichayapong, W. Pruthipunyaskul, W. Pattara-Atikom, and P. Chaovalit, "Social-based traffic information extraction and classification," in *Proc. 11th Int. Conf. ITS Telecommun.*, Aug. 2011, pp. 107–112.
- [8] M. Congosto, P. Basanta-Val, and L. Sanchez-Fernandez, "T-hoarder: A framework to process Twitter data streams," *J. Netw. Comput. Appl.*, vol. 83, pp. 28–39, Apr. 2017.
- [9] *Here Api Geocoder*. Accessed: Nov. 6, 2019. [Online]. Available: <https://developer.here.com/c/geocoding>
- [10] *Spain Electoral Roll Street Map, Spanish National Statistics Institute (INE)*. Accessed: Nov. 8, 2019. [Online]. Available: <https://www.ine.es/prodysser/callejero/>
- [11] N. R. Chopde and M. Nichat, "Landmark based shortest path detection by using A* and Haversine formula," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 1, no. 2, pp. 298–302, 2013.
- [12] 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.
- [13] D. Milne and I. H. Witten, "An open-source toolkit for mining Wikipedia," *Artif. Intell.*, vol. 194, pp. 222–239, Jan. 2013.
- [14] M. A. M. García, R. P. Rodríguez, and L. A. Rifón, "Leveraging Wikipedia knowledge to classify multilingual biomedical documents," *Artif. Intell. Med.*, vol. 88, pp. 37–57, Jun. 2018.
- [15] M. A. M. García, R. P. Rodríguez, and L. A. Rifón, "Wikipedia-based cross-language text classification," *Inf. Sci.*, vols. 406–407, pp. 12–28, Sep. 2017.
- [16] M. A. Mouriño-García, R. Pérez-Rodríguez, L. Anido-Rifón, and M. Vilares-Ferro, "Wikipedia-based hybrid document representation for textual news classification," *Soft Comput.*, vol. 22, no. 18, pp. 6047–6065, Sep. 2018.
- [17] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining Text Data*. New York, NY, USA: Springer-Verlag, 2012, pp. 163–222.

- [18] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.
- [19] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," 2015, *arXiv:1511.08630*. [Online]. Available: <http://arxiv.org/abs/1511.08630>
- [20] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" in *Proc. China Nat. Conf. Chin. Comput. Linguistics*. Kunming, China: Springer, 2019, pp. 194–206.
- [21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
- [22] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. Int. Joint Conf. AI*, Montreal, QC, Canada, Aug. 1995, vol. 14, no. 2, pp. 1137–1145.
- [23] M. Sahlgren and R. Cöster, "Using bag-of-concepts to improve the performance of support vector machines in text categorization," in *Proc. 20th Int. Conf. Comput. Linguistics (COLING)*. Stroudsburg, PA, USA: Association Computational Linguistics, 2004, p. 487.
- [24] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002, doi: [10.1145/505282.505283](https://doi.org/10.1145/505282.505283).
- [25] T. Sakaki, M. Okazaki, Y. Matsuo, and E. S. T. Users, "Real-time event detection by social sensors," in *Proc. 19th Int. Conf. World Wide Web*, 2013, pp. 851–860.
- [26] *Tweettronic*. Accessed: Sep. 15, 2020. [Online]. Available: <https://tweettronic.wordpress.com/>
- [27] *Web2express Digest*. Accessed: Sep. 15, 2020. [Online]. Available: <http://help.web2express.org/about-digest>
- [28] S. Vieweg and A. Hodges, "Rethinking context: Leveraging human and machine computation in disaster response," *Computer*, vol. 47, no. 4, pp. 22–27, Apr. 2014.
- [29] M. Demirbas, M. Ali Bayir, C. G. Akcora, Y. S. Yilmaz, and H. Ferhatosmanoglu, "Crowd-sourced sensing and collaboration using Twitter," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2010, pp. 1–9.
- [30] I. H. Li, W.-Y. Wang, Y. H. Chien, and N. H. Fang, "Autonomous ramp detection and climbing systems for tracked robot using Kinect sensor," *Int. J. Fuzzy Syst.*, vol. 15, no. 4, pp. 452–459, 2013.
- [31] B. Riveiro, H. González-Jorge, J. Martínez-Sánchez, L. Díaz-Vilariño, and P. Arias, "Automatic detection of zebra crossings from mobile LiDAR data," *Opt. Laser Technol.*, vol. 70, pp. 63–70, Jul. 2015.
- [32] R. R. Blázquez, M. M. Organero, and L. S. Fernández, "Evaluation of outlier detection algorithms for traffic congestion assessment in smart city traffic data from vehicle sensors," *Int. J. Heavy Vehicle Syst.*, vol. 25, nos. 3–4, pp. 308–321, 2018.
- [33] C. Prandi, S. Mirri, S. Ferretti, and P. Salomoni, "On the need of trustworthy sensing and crowdsourcing for urban accessibility in smart city," *ACM Trans. Internet Technol.*, vol. 18, no. 1, pp. 1–21, Dec. 2017.
- [34] C. Cardonha, D. Gallo, P. Avegliono, R. Herrmann, F. Koch, and S. Borger, "A crowdsourcing platform for the construction of accessibility maps," in *Proc. 10th Int. Cross-Disciplinary Conf. Web Accessibility (W4A)*, 2013, pp. 1–4.



MARIO SÁNCHEZ-ÁVILA received the B.S. degree in electronic communications engineering from the Universidad Complutense de Madrid, Spain, in 2017, and the M.Sc. degree in telecommunication engineering from the Universidad Carlos III de Madrid, Spain, in 2019.

From 2018 to 2019, he has been a Member of the Telematic Engineering Department, Universidad Carlos III de Madrid. He is currently a Project Engineer with Informatica El Corte Ingles (IECISA), where he works on projects related to electronic security and defense. His research interests include information extraction, social network analysis, semantic annotation, natural language processing, and machine learning.



MARCOS ANTONIO MOURIÑO-GARCÍA received the degree in telecommunications engineering and the Ph.D. degree from the University of Vigo, Spain, in 2013 and 2018, respectively.

From December 2013 to January 2019, he was Researcher with the Telematics Systems Engineering Group (GIST), Department of Telematics Engineering (DET), University of Vigo, where he developed his Ph.D. Thesis while participating in several national and international research projects about e-Health, e-Learning and active aging. From January 2019 to January 2020, he was a Visiting Professor with the Department of Telematic Engineering, Universidad Carlos III of Madrid. He currently works as a Data Scientist with Merlin Software for the Inditex Group. He has coauthored 20 scientific publications in national and international journals and conferences and book chapters. His main research include artificial intelligence, and specifically on its application to the automatic classification of text documents.



JESÚS A. FISTEUS received the M.Sc. degree in telecommunication engineering from the University of Vigo, Spain, and the Ph.D. degree in communication technologies from the Universidad Carlos III de Madrid.

He is currently an Assistant Professor with the Telematic Engineering Department, Universidad Carlos III de Madrid, Spain. He has participated in several international and national research projects. He is the author of publications in top international journals and conferences in the fields of technology-enhanced learning, semantic web, networking, and distributed systems.



LUIS SÁNCHEZ-FERNÁNDEZ received the M.Sc. degree in telecommunication engineering and the Ph.D. degree from the Technical University of Madrid, Spain, in 1992 and 1997, respectively.

In 1997, he joined the Universidad Carlos III de Madrid, where he currently occupies a position as Full Professor. He is a member of the Telematic Engineering Department, which he has headed, from 2009 to 2011 and from 2013 to 2015. He heads the Web Technologies Laboratory within the Telematics Applications and Services research Group (GAST). He has been the principal investigator of several national and international research projects. He has coauthored more than 100 scientific publications in national and international journals and conferences and book chapters. His current research interests include information extraction, social network analysis, smart cities, big data, and computational social choice.

• • •