information

Article

Extraction and Analysis of Social Networks Data to Detect Traffic Accidents

Nestor Suat-Rojas *, Camilo Gutierrez-Osorio and Cesar Pedraza

Programming Languages and Systems—PLaS Research Group, Department of Systems and Industrial Engineering, Universidad Nacional de Colombia, Bogota 999076, Colombia; cgutierrez@unal.edu.co (C.G.-O.); capedrazab@unal.edu.co (C.P.)
* Correspondence: nsuat@unal.edu.co

Abstract: Traffic accident detection is an important strategy governments can use to implement policies intended to reduce accidents. They usually use techniques such as image processing, RFID devices, among others. Social network mining has emerged as a low-cost alternative. However, social networks come with several challenges such as informal language and misspellings. This paper proposes a method to extract traffic accident data from Twitter in Spanish. The method consists of four phases. The first phase establishes the data collection mechanisms. The second consists of vectorially representing the messages and classifying them as accidents or non-accidents. The third phase uses named entity recognition techniques to detect the location. In the fourth phase, locations pass through a geocoder that returns their geographic coordinates. This method was applied to Bogota city and the data on Twitter were compared with the official traffic information source; comparisons showed some influence of Twitter on the commercial and industrial area of the city. The results reveal how effective the information on accidents reported on Twitter can be. It should therefore be considered as a source of information that may complement existing detection methods.

Keywords: intelligent transportation system; social media; traffic accident; social sensors; natural language processing; machine learning; text mining; classification; named entity recognition

check for updates

Citation: Suat-Rojas, N.; Gutierrez-Osorio, C.; Pedraza, C. Extraction and Analysis of Social Networks Data to Detect Traffic Accidents. *Information* **2022**, *13*, 26. https://doi.org/10.3390/info13010026

Academic Editor: Barbara Guidi

Received: 20 November 2021
Accepted: 4 January 2022
Published: 10 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

The population and economic growth of cities are the main causes of the increase in the number of vehicles [1]. This increase is reflected in a greater number of traffic accidents; according to the National Road Safety Agency [2], in 2016 there were 7000 deaths and 45,000 injuries reported in Colombia, the highest number since 2000. The identification of the key factors of traffic accidents in the city provides government officials with a tool to build policies and actions to reduce these types of incidents. The detection of road incidents makes it possible to determine peak hours, the most common regions and segments, and other influencing factors such as social events [3], road infrastructure [4], weather and lighting, among others.

Some proposals for traffic detection and monitoring suggest using physical devices such as video cameras [5], loop inductors [6,7] and RFID readers [8]. However, these devices are required to be fixed on major road segments, thus reducing coverage; other common problems that reduce traffic forecasting are maintenance costs and accuracy errors due to weather conditions. This has motivated researchers to study the effectiveness of other available secondary sources of information—such as social media—for the detection of traffic incidents, expanding the coverage in the city, involving all road safety stakeholders [9,10] and offering the possibility of analyzing other factors such as mass events and road conditions [11]. These authors propose methodologies to extract data relevant to traffic incidents from social networks, using machine learning and text mining techniques [10,12,13]. These techniques allow monitoring traffic incidents using the

resources wisely, so that local governments and researchers can gain control in these types of accidents.

This paper proposes an approach for extracting accident-related data using Twitter Spanish posts. It consists of a study conducted from October 2018 to July 2019 in the city of Bogota (Colombia). It is divided into four phases: data collection, classification of accident-related tweets, entity recognition for location and time extraction, and geolocation of reported accidents. The first phase contemplates the mechanisms for data collection on Twitter and the construction of a labeled dataset using crowdsourcing data labeling (It is the process of data labeling in which a group of people collaborate) with the collaboration of 30 people and a selection mechanism of three votes per tweet; the tweets were collected through Twitter APIs: Search API and Stream API. A collection process was designed taking into account the tweet search by keywords and the tweet collection from official accounts. The second phase implements an automatic classification method using the Support Vector Machine (SVM) supervised learning algorithm, trained with 3804 tweets, where half of them are related to traffic accidents. This method allows filtering tweets relevant to traffic accidents and non-relevant tweets in Spanish; in this phase, a doc2vec embedding model is also built with one million tweets. In the third phase, an entity recognition technique is used to extract the location and time of the accident from the text. For this task, the Spacy library was implemented, re-training the es_core_news_lg model (Authors on https://explosion.ai/, accessed on 20 November 2021) with our own dataset consisting of 1340 tweets using the IOB format. In the fourth phase, after extracting the location, it was processed by a geocoder used to return the geographic coordinates of the address concerned.

The remainder of this paper is organized as follows: Section 2 provides a literature review on accident detection using social media. Section 3 exposes the methodology and architecture designed for tweet processing and traffic accident data extraction. The description of the consolidated data set, as well as the description of the experiments design, is detailed in the results of Section 4. That section also evaluates the proposed collection method by comparing the data extracted from other official sources. Section 5 presents the discussions and conclusions.

2. Literature Review

2.1. Analysis of Traffic Accidents

The identification of the key factors of traffic accidents provides government officials with a guide to build policies and actions to reduce these incidents. In the literature, there are studies related to the detection and analysis of anomalies and traffic accidents, using data collected by fixed devices and available physical infrastructure such as video cameras [5], loop inductors [6,7], RFID readers [8,14] and GPS information provided by public service vehicles [15] or private vehicles [16]. Kwak and Kho [17] performed a logistic regression analysis using data collected by loop inductors, identifying the most significant traffic variables that cause traffic accidents, such as unsafe speeds, vehicles following too closely, and weather conditions. Zuo et al. [16] used the Chinese geopositioning system Beidou (BDS) to analyze the traceability of a vehicle moments right before the accident and identify the driving behavior prior to the incident. Liu et al. [18] used historical GPS data from cabs and buses to model normal traffic patterns, and then detect anomalies when observed traffic deviates from what the system predicts. However, these devices require a cost of implementation that cities cannot afford, thus limiting coverage since they must be installed in fixed areas; other common problems that reduce traffic forecasting are maintenance costs and accuracy errors due to weather conditions.

2.2. Information on Traffic Accidents Posted on Social Media

Extracting traffic information from a single source of information is ineffective for accident analysis; most of the physical devices that are commonly used have limitations related to coverage, costs and accuracy. This has motivated researchers to study the ef-

fectiveness of social media as information sensors available for the detection of traffic incidents, extending the coverage in the city, involving all road safety stakeholders [9,10], and offering the possibility of analyzing other factors such as mass events and road conditions [11]. Given its *API* (Application Programming Interface or API; it is used as a layer for accessing the data of an executing application) is widely developed and available, recent studies on social media mining to extract traffic information has focused on Twitter as a data source. Wang et al. [3] validated the effectiveness of Twitter by comparing it with data collected from GPS on public transport buses. Similarly, Gu et al. [11] also demonstrated a first approximation by correlating traffic flow with social events and road infrastructure. Additionally, Zhang et al. [13] compared accident reports extracted from Twitter with data obtained from loop inductors and police databases, obtaining similar results. However, not all accidents that occur are reported in social networks. Therefore, Wang et al. [3] concluded that social media should be considered as complementary sources of information that expand coverage of physical devices on secondary and arterial roads.

Several authors have proposed methodologies to extract information on traffic incidents from social networks, using machine learning and text mining techniques [10,12,13]. These techniques allow monitoring traffic incidents using the resources wisely, so that local governments and researchers can gain control in these types of accidents. Given the variety of techniques for data mining and vehicular traffic monitoring using physical sensors and social networks, studies have been undertaken to combine these data sources or heterogeneous data, using deep learning techniques [19], matrix factorization [3] and designing architectures to combine these sources [20].

2.3. Methods for Incident Extraction from Social Media

Twitter allows to easily extract information given that most profiles are public. Additionally, other methodologies have been used to extract data relevant to traffic incidents from Twitter [11,12,21–23], contemplating different phases for tweets acquisition; preprocessing; classification and geolocation to extract incident coordinates. At each phase, natural language processing methods are used to solve the various challenges posed by the language used in social networks, such as spelling errors, use of non-formal language, abbreviations and new expressions that arise with the evolution of language, word ambiguities, as well as noise present in posts [10,11,13].

2.3.1. Automatic Classification of Tweets Related to Accidents

There are methods proposed in the literature to classify tweets related to traffic events [12,21,24] and traffic accidents [25,26]. These studies employ an automatic classifier that uses machine learning algorithms [11] and natural language processing techniques [27,28]; the phases include tweet preprocessing, feature extraction and classification models.

- Preprocessing: In this phase, the posts are received, cleaned and normalized, dividing the sentences into words or segments called tokens (tokenization). Web links or URLs, user mentions (@), empty words or *stopwords*, special characters and numbers are eliminated. Finally, the words are reduced according to their lemma (*lemmatization*) or their stem (*stemming*).
- Feature extraction: The sentences generated in the previous step are represented as features or vectors to improve the performance of the mathematical models. In the literature, the *TD-IDF* technique has been implemented, which represents the frequency of the words in the tweet with respect to the number of tweets in the corpus [27,29]. Another technique being used is *word2vec*, which is a trained neural network that generates smaller semantic arrays taking into account the context of the words within the sentence [24,30].
- Automatic classification: From the results of the previous phase, a machine learning model is built to determine whether a tweet is related to an incident or not; some algorithms used in the literature are *naive bayes* [11,25], *support vector machine* [21,23,28] and *convolutional neural networks* [30]. Each model has different parameters that must

be configured to obtain better results; these parameters are searched through an optimized intensive search, such as the grid search method.

2.3.2. Named Entity Recognition

Entity recognition is a natural language processing task that extracts information from text, such as location, organization, time, person, among others. Different techniques for extraction in social media have been explored in the literature such as *rules-based* [27] and *sequence labeling* [31,32], the latter being state-of-the-art and without requiring the construction of exhaustive dictionaries or *gazetteers*—it is based on a manually labeled dataset that allows to train, learn and generalize on new data. In the case of traffic accident reports, the aim is to extract the *location* and *time* entities. In this case, some authors [11,12] used a similar methodology, starting by pre-processing tweets and selecting a location recognition technique, also known as *geoparsing* [33].

- *Preprocessing*. Some URLs, emoticons and special characters are deleted. The @usernames and #hashtags may contain location information and some authors do not eliminate them. These expressions are generally formed by two or three words without spaces, so a method is applied to expand them [34,35]. Furthermore, spelling errors and abbreviations commonly used in social media are corrected [33]. In order to recognize entities, *stopwords* are not eliminated because these words are key to understand the context of the following word and avoid ambiguities.
- *Entity recognition*. There are different techniques that can be employed to extract location entities in tweets. In general, they are classified into two types: *based-rules* and *sequence labeling*. Some authors use a hybrid of both techniques [11,33,35–37]. The main difference lies in the use of resources external to the text content itself, such as *based-rules* techniques that build dictionaries of terms, toponymies and metonymies; local *Gazetteers*; a set of rules or a list of regular expressions. On the other hand, *Sequence Labeling* techniques use supervised machine learning methods, using only a set of labeled features and data from text, thus training, learning and generalizing on unseen data. The best performing models for the sequence labeling task in social networks are *CRF* [37], *BiLSTM* [31] and *BiLSTM+CRF* [32].

2.3.3. Geocoding

Once extracted, the location entities are processed through a *geocoder* that returns a geographic coordinate approximating the location reference; there are external resources that can be used such as *Google*, *OpenStreetMap*, *Geonames*, *DBPedia*, among others. Some authors also built geocoders that search for matches in a *gazetteer* using search algorithms such as *string match* or *fuzzy string match* [11].

3. Materials and Methods

Our approach is divided into four phases for the collection, classification and recognition of entities in spanish tweets. Figure 1 shows the design. The first phase contemplated the mechanisms for the collection of Twitter data and the construction of a labeled dataset using *crowdsourcing data labeling*. The second phase implemented an automatic classification model that allowed filtering tweets related to traffic accidents and non-related tweets. In the third phase, entity recognition techniques were used to allow the extraction of location and time from the text. In the fourth phase, after extracting the location, it was processed by a geocoder used to return the geographic coordinates of the address concerned.

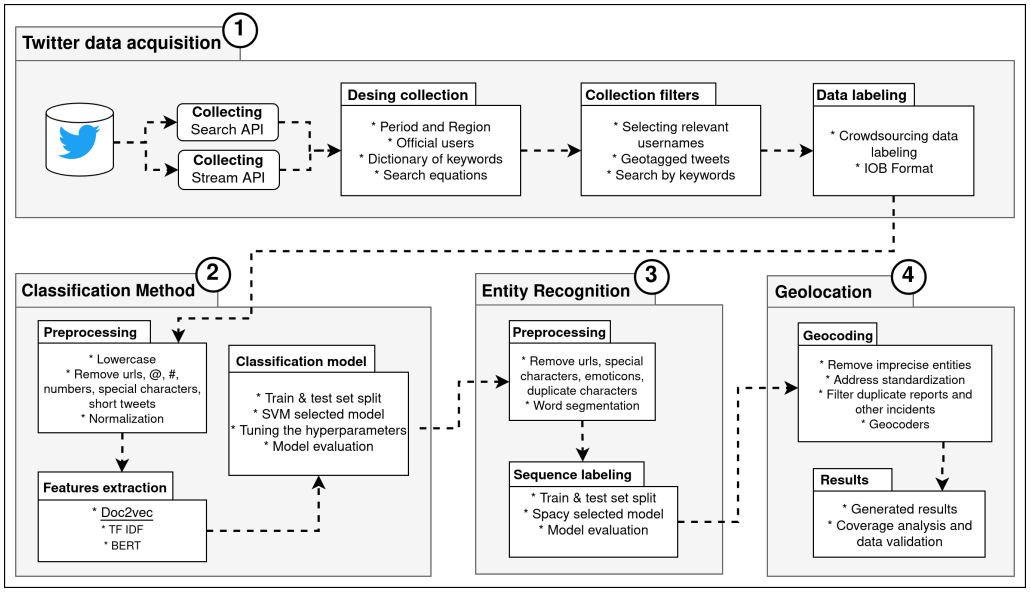

Figure 1. Proposed methodology for automatic detection of traffic accidents on Twitter (**1**) Twitter data acquisition; (**2**) Classification method to automatically filter tweets on accident reports; (**3**) Extraction of location and time entities from the traffic accident report; (**4**) and geolocation of accidents.

3.1. Twitter Data Acquisition

Tweets are collected with two resources provided by the Twitter API. The first resource is called Search API, which allows searching by keywords and tweets of selected user accounts. The second resource is the Stream API service, which receives 1% of the tweets posted in a specific geographic region in real time. Given the limitations of the free API, and with the purpose of collecting as many tweets as possible, four types of collection filters are built that combine both resources mentioned above.

3.1.1. Design Collection

Before starting the data collection process with the Twitter API, some search criteria must be previously defined, such as building a keyword dictionary, establishing relevant user accounts and defining the geographic region and collection period.

- *Period and region*. The tweets in Spanish correspond to the city of Bogota, Colombia and were collected for ten months: from October to December 2018 and from January to July 2019.
- *Official users*. Given the importance of traffic reports in the city of Bogota, the official user accounts shown in Table 1 are selected. In this table the user profiles represent some important sector such as @BogotaTransito which belongs to Bogota city transit authority, @CIVICOSBOG and @RedapBogota belong to civic organizations, @Cititv which is a news channel account and @transmilenio and @rutassitp belong to public transport services.
- *Keywords dictionary*. The Twitter API enables to extract tweets based on a keyword match. In order to determine the list of keywords, the words (sequences of words called n-grams) with the highest occurrence among hand-picked documents, reports and tweets related to traffic accidents are manually extracted.
- *Search equations.* Some search equations are created based on the word dictionary to optimize the extraction. As seen in Table 2, these equations include the word exceptions to discard irrelevant tweets; the "-" sign indicates these words should be excluded. Some user accounts or hashtags are also included in the search.

Table 1. Usernames of local government and mass media official accounts.

Twitter Usernames			
@BogotaTransito	@Citytv	@RedapBogota	@WazeTrafficBog
@CIVICOSBOG	@rutassitp	@SectorMovilidad	@UMVbogota
@idubogota	@transmilenio	@IDIGER	

Table 2. Search equations defined for tweets in Spanish and Bogota.

Search Equations
(“accidente” OR “choque” OR “incidente vial” OR “incidente” OR “choque entre”) -RT -“plan de choque”
(“atropello” OR “tráfico” OR “trafico” OR “tránsito” OR “transito” OR “#trafico” OR “#traficobogota” OR “sitp” OR “transmilenio”) -RT

3.1.2. Collection Filters

Using the two tools provided by Twitter to access tweets—Stream API and Search API—four types of filters are defined as different collection methods.

1. *Stream Bogotá*. Query to extract tweets posted in Bogota with Stream API. The free version only extracts 1% of the tweets. Furthermore, it only provides the tweets with the geotag option enabled or those posted from cell phones with GPS enabled.
2. *Stream Follow/Timeline User*. Tweets collected using Twitter Stream API. One or several users are selected; and an automatic download is started the moment the user posts a tweet or someone tags him/her in a comment.
3. *Search Token*. Tweets are collected using Twitter Search API. In this case, the search equations defined in Table 2 are used; additionally, the results are filtered for Bogotá and Spanish.
4. *Search Timeline User*. Tweets are collected from five selected users from Table 1 using Twitter Search API. (@BogotaTransito, @Citytv, @RedapBogota, @WazeTrafficBog, @CIVICOSBOG). The tweets posted in their timelines are downloaded, and the API allows to extract the user’s tweet history.

3.1.3. Data Labeling

Once the first tweets are collected, they are used to build an initial dataset for training and evaluating the machine learning models in the following phases. For this task, the dataset is labeled to express the ground truth that the models are expected to learn. Depending on the phase, the labeling process and format may change, as shown below.

Crowdsourcing Data Labeling for Classification

For training the tweet classification model, a collaborative labeling strategy was designed. Here, 30 people labeled data according to the instructions given. Each participant had to evaluate a tweet to manually classify it into one of three categories defined as: traffic accident related, unrelated and do not know/no response. In this process, 22,582 tweets were randomly selected from October to December 2018. As a result, 3505 tweets were labeled as “traffic accident related”. Each tweet was evaluated by three participants. The correct label was selected by voting; the 3 people must agree on the selected label, otherwise the tweet was excluded from training. This process took a month and required the development and deployment of a web application, which was named Tagenta (Source code available on https://github.com/solofruad/traffic-accidents/tree/master/accident-tagger, accessed on 20 November 2021).

Sequential Data Tagging

In order to train and evaluate the entity recognition model, a sample of 1340 tweets was selected. They had to include accident-related information such as **location**, like places and addresses; and **time**, like "ayer/yesterday" or "esta mañana/this morning". Subsequently, this dataset was manually labeled using the *IOB (Inside-outside-beginning)* format. The labeling tool called Brat Annotation Tools was used for this task, as shown in Figure 2, with this tool we can select the segment of tweets that reference the selected tags. The labels defined are Location, which refers to the location of the report; and Time, which refers to the time or date of the incident. Accordingly, 5 labels were generated: *B-loc*, *I-loc*, *B-time*, *I-time* and *O*. The *O* label refers to *Others*.

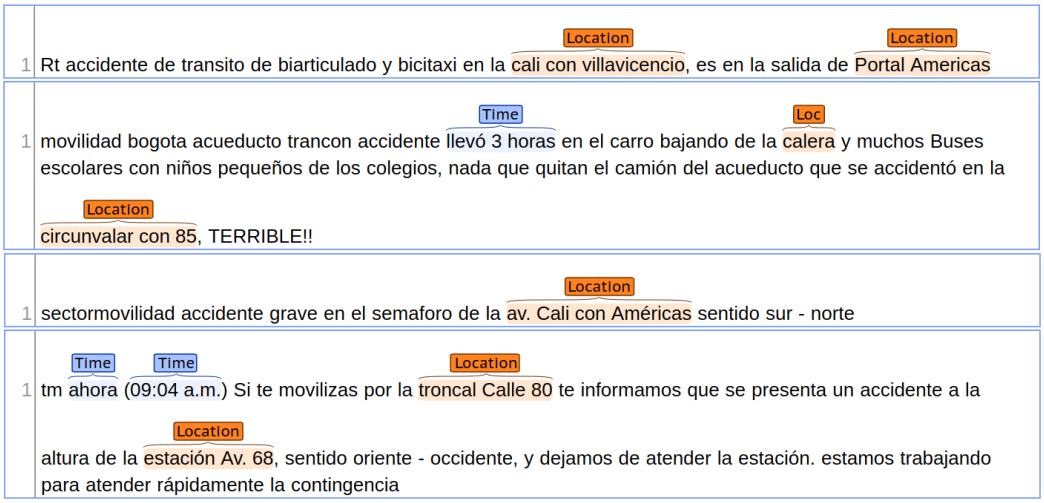

Figure 2. Manual labeling using Brat Annotation Tools.

3.2. Classification Method

This phase was divided into three sections. First, the tweets go through a normalization and cleaning process; second, they are transformed into a feature vector representation; and finally, an automatic classification model categorizes them as accident-related or -unrelated. In each process, different techniques proposed in the literature for processing social media posts were tested.

3.2.1. Preprocessing

Due to their nature, social media posts often employ informal language, emoticons, user mentions and topic labels such as hashtags, among others. In order to identify a clear pattern in traffic accident reports, their content must be cleaned and normalized. The actions performed on the tweet contents were as follows.

A Conversion to lowercase: Since word dictionaries are case sensitive, all words in tweets were shifted to lowercase.

B Cleaning of non-alphabetic characters: Special characters, e-mail addresses, URLs, including @ and # symbols that are popular on Twitter were removed.

C Normalization with lemmatization: In Spanish, words have variations and are conjugated in different forms. To find a more generalized pattern, a normalization process was applied to tweets, which consisted of reducing the word to its original lemma. Another form of normalization is stemming which, instead of reducing it to its lemma, it reduces the word to its stem. It consists of removing and replacing suffixes from the word stem. In the following sections, the performance of the proposed method is evaluated by comparing lemmatization and stemming.

D Tweets whose word count or token length was fewer than three after performing the steps above are discarded.

Use of stopwords. Stopwords are words consisting of articles, pronouns, prepositions, among others. These words are considered worthless or empty because they are usually in both related and unrelated posts. Therefore, they do not add value to discriminate one type of publication from another. Some feature extraction models (Section 3.2.2) can take care of these stopwords without having to remove them first; e.g., the *TF-IDF* algorithm can ignore those words that are most frequently repeated in the corpus. Other algorithms based on neural networks—such as *doc2vec* and *BERT*—do not remove stopwords because they are key to give context to each sentence. If these words are cleaned up using a generic stopword dictionary, some words relevant for the context of traffic accident reports may get lost, such as the prepositions "entre/between", and "con/with".

3.2.2. Features Extraction

After applying the cleaning and normalization process, the resulting text content is represented as a vector space, commonly known as Embedding. The technique selected in this study is doc2vec [38], which has shown good results with the processing of posts in social media [30,32,39]. An example is the work of Pereira et al. [24] who implemented doc2vec for the classification of transit-related tweets in Portuguese from the cities of São Paulo and Rio de Janeiro.

Doc2vec is an embedding method proposed by Le & Mikolov et al. [38] that extends the implementation of word2vec [40] to a sequence of words or paragraphs. It consists of a single hidden layer neural network that aims to learn the context of the sentences and words used without supervision, generating a vector representation that takes into account the semantics of the sentences. Doc2vec has two different methods that can be used. In this case, ***DBOW*** or ***Distributed Bag of Words*** was selected because of its remarkable performance in social media in recent studies such as those by Okur et al. [39] and Agilar et al. [32]. DBOW predicts the representation of the context word according to a target word.

In addition, to evaluate the performance of doc2vec on Spanish tweets, other embedding techniques used in the literature such as *TF-IDF* and *BERT* were tested.

TF-IDF. This model calculates a representation taking into account the frequency of occurrence of a word in the same tweet and in the whole corpus collection.

BERT. A pre-trained embedding method is used to extract the vector representation of a text. It was launched by Google at the end of 2018 [41] based on the idea of transformers or deep neural networks consisting of multilayer encoders. BERT is available in two versions, a 12-layer version for the base model and a 24-layer version for the extended model. For this study, the 12-layer model of Cañete et al. [42] was used, which was trained using a Spanish corpus and the case-sensitive version.

3.2.3. Classification Model

The last process of this phase is the construction of a machine learning model to perform a binary classification, predicting whether a tweet is related or not to a traffic accident. The model selected is the **Support Vector Machine** or **SVM**, because of its implementation in several studies related to traffic tweets that have demonstrated its effectiveness compared to other models [21,23,24,26,28,29]. Similarly, to compare the performance of SVM with tweets in Spanish, other models used in the literature such as Naive Bayes (NB), Random Forest (RF) and Neural Network (NN) were evaluated, the latter being selected as a recommendation of the BERT authors for classification tasks. The results obtained from this comparison are presented in the following experiments and results subsections (Section 4).

3.3. Entity Recognition

This phase allowed to extract *Location*, which refers to the location of the report; and *Time*, which refers to the time or date of the incident. Here, another machine learning model was applied independently from the previous one. Hence, this phase was divided into

two sections: preprocessing and training of the sequence labeling model for named entity recognition tasks.

3.3.1. Preprocessing

Similar to the classification model, to improve the performance of the Named Entity Recognition (NER) model—and given the informal language used in social media—each tweet was cleaned and normalized. However, preprocessing is different for NER. Special characters and URLs are removed, but @usernames and #hashtags may contain location information, thus, word segmentation is applied to them [43]. Preprocessing is as follows:

- Remove unnecessary ASCII codes, URLs and line breaks.
- Remove special characters and emoticons, except for punctuation marks and accents.
- **Delete letters or punctuation marks that repeat consecutively.** If a letter is repeated more than two times in a row, the other letters are deleted. For example, "goooool" is replaced by "gol" (goal in English). In the case of symbols, if they are repeated more than three times in a row, they are deleted.
- **Word segmentation for #hashtags and @usernames.** As suggested by Malmasi & Dras [35], these expressions may contain location information. However, they are usually a combination of several words. Therefore, the word segmentation model proposed by Norvig [43] was used.

Word Segmentation in Social Media

In social networks, users often use expressions known as #hashtags. They highlight a topic and can be identified or classified later by other users. These expressions are usually a combination of words and numbers. Hashtags (#) may contain location information [35], as well as user mentions (@) that sometimes contain the name of a town, neighborhood or point of interest such as buildings or parks. For this reason, this content was not discarded and a word segmentation model proposed by Norvig [43] was used to separate those words.

A *Dataset*. A corpus with 400 K unigrams was built using the Spanish dataset of Cañete et al. [42] with about three billion tokens and the in-house dataset extracted from Twitter with 76 million tokens.

B *Cleaning and normalization*. The same preprocessing mentioned above was applied. In this case, the accent was removed from the words so they did not have any accent. Each text was divided into unigrams to create a dictionary.

C *Naive Bayes-based language model*. With the dataset, a language model was created, which predicted several word segmentations and the most probable one was selected.

Some word segmentation results can be seen in Table 3. For example, the expressions #puentearanda, #avenida68 and @CorferiasBogota expand, respectively, into "puente aranda", "avenida 68" and "corferias bogota" which are places of interest in Bogota. The segmenter recognized proper nouns such as soccer teams, points of interest and common abbreviations mentioned in social networks.

Table 3. Results of the word segmenter with tweets in Spanish.

#hashtag or @username	Results
#puentearanda	puente aranda
#avenida68	avenida 68
@CorferiasBogota	corferias bogota

3.3.2. Sequence Labeling

For entity recognition from text, a machine learning model based on sequence labeling was used. The objective was to train a model that recognizes the location and time entities mentioned in tweet posts. The model selected is available in **Spacy** (Available on https://spacy.io/, accessed on 20 November 2021) [44], a state-of-the-art library

used in Python for Natural Language Processing tasks in several languages. Spacy has a pre-trained deep learning model in Spanish called es_core_news_lg (Authors on https://explosion.ai/, accessed on 20 November 2021). This model was re-trained with our set of labeled tweets. Consequently, 500 iterations were used for training, as suggested in the library documentation.

To assess Spacy's performance with Spanish tweets, it was compared with the following machine learning models used in the literature.

Conditional Random Fields (CRF). A model based on local features that are extracted from words and their neighbors. For this implementation with tweets, some features were taken into account based on previous studies conducted by Taufik et al. [45], Okur et al. [39] and García-Pablos et al. [46].

Bidirectional Long Short-Term Memory (BiLSTM). Unlike CRF, it does not require local features. The bidirectional architecture consists of connecting two different layers of LSTM recurrent neural networks, one to analyze the sequence from left to right and the other to analyze the sequence from right to left. Therefore, it takes into account the context of the word according to the neighbors. This model was implemented according to the recommendations of Peres et al. [31] and Aguilar et al. [32].

BiLSTM + CRF. Aguilar et al. [32] suggested adding an output layer using a CRF model, thus transferring what is learned by the BiLSTM model to the last layer.

3.4. Geolocation

Once the locations were extracted in the previous model, they went through a geocoding process that returned the geocoordinates. This information was then used to validate the integrity of the data in Twitter through a coverage analysis. However, since informal language is used, it is recommended to make some previous steps, such as removing tweets with insufficient and inaccurate location information and applying an address standardization process. This section is divided into four tasks as described below.

A *Remove tweets with inaccurate location entities*. Some tweets make reference to more than one location or include vague references to the location of the incident without specifying further details, which makes it difficult to geolocate the coordinates. For this reason, it was decided to apply the following two rules to filter these tweets. The underlined text refers to the location entities detected by the method.

- Discard tweets with fewer than four words in the detected location. For example, "*Accidente en Calle 22 trafico bogota*, the entity "*calle 22*" does not provide sufficient information.
- Discard tweets with more than four recognized location entities. Tweets mentioning different locations or addresses in the report. For example, -"*Calle 34 trancada al Oriente desde la calle 26 hasta la carrera 13 -choque en la Glorieta de la Avenida Primero de Mayo con carrera 68*, is a tweet with two traffic reports in different locations.

B *Address Standardization*. The addresses or locations detected in the tweets lack a formal language, thus limiting the effectiveness of the geocoders. Some drawbacks are the use of abbreviations, different toponymies for the same place and lack of precision in the address or incomplete place names. These problems motivated to use a method to enrich and standardize the locations detected in the tweets messages. For this reason, the Libpostal library was used (repository available on https://github.com/openvenues/libpostal, accessed on 20 November 2021). This library allowed to adapt the dictionaries of words, toponymies, abbreviations and other modifications made to adapt to the particularities of Bogota. Finally, Libpostal can transform a location recognized as "*cl 72 * cra 76*" → "*BOGOTA AVENIDA CALLE 72 CARRERA 76*".

C *Filtering duplicate reports and other incidents*. To reduce the cost of using paid geocoders such as Google Maps API, which charge per query, duplicate reports and noise present in Twitter were first removed. The objective was to extract additional information on Twitter and not duplicate existing information. At this point, tweets with similar

locations—extracted in the previous step—and occurring at the same address in a 1-h time window (before and after) were removed. In addition, to filter tweets from other events or incidents, a final selection was made by matching keywords related to accidents, thus eliminating non-relevant reports.

D *Geocoders*. Once the addresses were standardized, a geocoder returned a geographic coordinate. An app called *batch geocode* (Available on https://github.com/GISforHealth/batch_geocode accessed on 20 November 2021 (GISforHealth's Github repository)) was used, which combines resources available on *Google*, *OpenStreetMap* and *Geonames*. Up to four results were fetched per resource and the geocoding tool automatically assigned a coordinate if all points fall within a zone of influence.

4. Results

4.1. Case Study: Traffic Accident Report in Bogotá D.C. (Colombia)

This study collected and processed tweets on traffic accidents in Bogotá D.C. (Colombia) during a ten-month period, from October to December 2018 and January to July 2019. The city of Bogotá has a total area of 1775 km² divided into 20 administrative divisions or localities, of which 307 km² and 19 localities correspond to the urban area; the city has a population of 7 million inhabitants. Accidents reported in the official database of the Secretariat of Mobility for the same period amount to 25,299 incidents, most of which were reported in the urban area as shown in Figure 3.

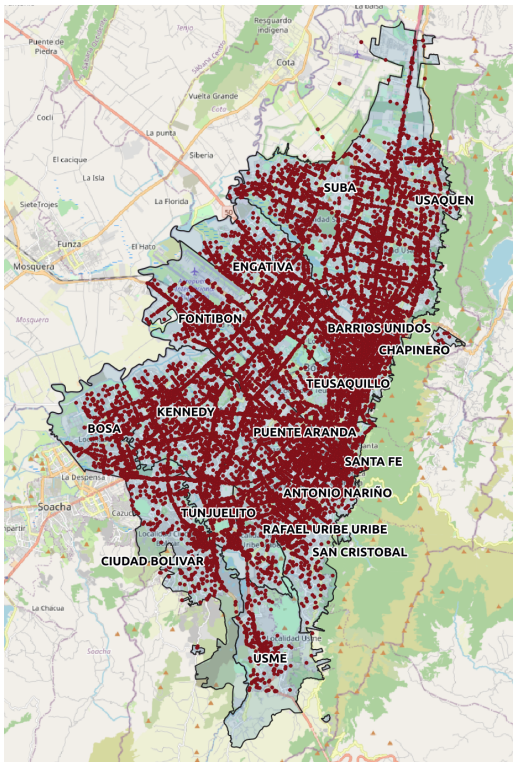

Figure 3. Location of traffic accidents that occurred from October 2018 to July 2019.

4.2. Data

The data used for this paper were collected using the mechanism defined in Section 3.1 and the collection filters in Section 3.1.2 for the ten-month period between October 2018 and July 2019. A total of 4,973,900 tweets were collected. Table 4 shows the amount extracted according to the collection filter, the designed filter that extracted the most amount of tweets is "Stream Bogotá", which represents the majority of the dataset to be analyzed with 4,027,313 records. Table 5 shows a sample of the extracted tweets that are accident reports containing the information of the location of the event, in many occasions with

good description of the location, others with a vague reference, and also in some cases containing more than one accident report.

Table 4. Number of tweets collected by filter.

Collection Filter	# Collected Tweets
Stream Bogotá	4,027,313
Stream Follow/Timeline User	574,816
Search Token	271,153
Search Timeline User	100,618

Table 5. Tweets collected in Bogotá.

Tweets	Description
Hueco causa accidentalidad Cra. 68c #10-16 sur, Bogotá	Accident
Justo ahora 1:22 p.m. en 21 angeles Av Suba gratamira (calle 145 Av Suba) complicaciones viales por accidente @BogotaTransito @ALCALDIASUBA11 @SectorMovilidad	Accident
Semáforos de la Carrera 24 con Calle 9 en amarillo intermitente, Tanto por la calle como por la carrera con riesgo de incidente vehicular	No Accident
Incidente vial entre bus y un motociclista en la calle 86a con carrera 111a. Unidad de @TransitoBta y asignadas.	@BogotaTransito
• en la avenida Primero de Mayo con carrera 69 en sentido occidente-oriente chocan un taxi y una motocicletaen • la avenida de La Esperanza con carrera 68 A en sentido occidente-oriente chocan un vehículo particular y una camioneta	Two different reports in the same tweet
#ArribaBogotá Por culpa de este hueco en la calle 27sur, una mujer sufrió un grave accidente de tránsito.	Vague location

4.2.1. Data for Classification Method

As mentioned in Section 3.1.3, once the first tweets were collected, the dataset was built to train and evaluate the machine learning models. Following the crowdsourcing data labeling process (Section 3.1.3), 3505 tweets labeled as related to traffic accidents between October to December 2018 were obtained. However, not all of these tweets were used for training the classification model. Most of these posts—2898 tweets—were from the official accounts of *@BogotaTransito* and *@rutasstip* and contained tweets with the same format that included the sentence "Traffic incident between". For this reason, only a fraction of the tweets from these official accounts was selected. Hence, their tweets only represent 30% of the selected sample and the other users represent 70%. This was made in order to avoid a bias in the set of positive data or data related to traffic accidents.

Considering the above, the dataset for the classification model contains **3804 tweets**, where 1902 are related to traffic accident reports (TA, positive class) and 1902 are unrelated (NTA, negative class). For performance evaluation, the dataset was divided in two ways, (1) cross validation with k equal to 10 for evaluating the SVM, RF, and NB models; and (2) in the case of the NN, the dataset was divided into 70% (2662 tweets) for training and 30% (1142 tweets) for testing.

4.2.2. Data for Entity Recognition

For the entity recognition model training, a sample of the filtered tweets resulting from the previous classification phase was taken. **1340 tweets** were extracted, where 800 are from "unofficial" users, almost 60% of the sample. These tweets were user reports on traffic incident occurred in Bogota from October 2018 to July 2019, including other tweets that contained some location references such as reports on the state of road infrastructure; some

tweets from the years 2016 and 2017 were also included. Although these posts were not related to accidents per se, they were selected because they contained location information. The purpose was to train a model that would recognize these entities, because a classifier of accident-related tweets was previously created. Additionally, the dataset was split, reserving *1072 tweets for training* and *268 for evaluation*.

In the extracted tweets there are posts with many vague references to addresses such as "accident at 10 and 15" without specifying a street or avenue or a point of interest. Furthermore, there were tweets that referred to more than one location in the same message, mentioning two events in different locations. Some of these cases are detailed in Table 5.

The labels defined are Location, which refers to the location of the report; and Time, which refers to the time or date of the incident. Accordingly, 5 labels were generated: B-loc, I-loc, B-time, I-time and O. The O label refers to "Others". Table 6 describes the number of entities in both datasets, respectively. In this table it can be seen that the dataset contains 1831 locations (B-loc) and 164 time-related tags (B-time), the other tags such as I-loc and I-time are their descriptions.

Table 6. Number of tokens per label and dataset.

	B-Loc	**I-Loc**	**B-Time**	**I-Time**	**O**
Trainset	1462	3768	131	112	20,242
Testset	369	893	33	33	5038
Total	1831	4661	164	145	25,280

4.3. Experiment

The objective was to compare the efficiency of each preprocessing method in Section 3.2.3, each embedding method—TF-IDF, Doc2vec and BERT—, and each machine learning algorithm in Section 3.2.3. The metrics used for this comparison were **Accuracy, Recall, Precision** and **F1 score**. These results allowed selecting the best classification technique for tweets related to traffic accidents.

The parameters used for TFIDF, SVM and RF were found through a hyperparameter search using the grid search method of the *Sklearn* library available in Python. For the process of the fitting of the parameters, the grid search method finds the optimal values in the models through an exhaustive exploration, here we can evaluate the combined performance of the embedding and classification models, and in this way find the best configuration for both to function properly. The Sklearn library makes this exploration easier by assigning at the beginning a dictionary of different values that the parameters will take, then it performs a combination of all these until the best one is found. Five evaluations were performed for each combination experiment, where in each evaluation the data set is partitioned differently, separating one set for training and another set for testing, this process is known as the cross validation strategy with parameter k equal to 5.

4.3.1. Evaluation Metrics

To evaluate the performance of the proposed machine learning models, a set of metrics per class called Accuracy, Recall, Precision and F1-score were used. These measures are widely used in related studies to evaluate classification models and sequence labeling [21,26,27,30]. As these metrics are for a given class with l label, the following indexes were calculated first: True Positive (TP) are the number of instances of l label that were correctly classified with l label; and True Negative (TN) are the number of instances with any other label except l that were correctly classified. On the other hand, False Positives (FP) are the number of instances with any other label except l that were misclassified with l label; and False Negatives (FN) are the number of instances with l label that were misclassified with any other label except l.

Accuracy (acc) indicates the fraction of correctly classified labels. It is the division of correctly classified labels by the total number of labels (1).

Precision (P) indicates the fraction of instances with the l label that were correctly classified out of all the instances classified in that class. Precision measures the correctness of the classifier (2).

Recall (R) indicates the fraction of instances with *l* label that were correctly classified out of the entire population of instances that actually belong to that class. Recall measures the effectiveness of the classifier (3).

F1-score (F1) is the harmonic mean between precision and recall (4).

$$acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{4}$$

Classification

To compare the different classification models, the aforementioned metrics were defined. In this case, Traffic Accident (TA) reports were evaluated as label l_1 and Non-Traffic Accident (NTA) reports as l_2 label, respectively.

Sequence Labeling

The metrics defined for this experiment were F1-score, Recall and Precision for each class or label. At the end, a single measure was calculated for the model using the weighted, F1-weighted, R-weighted and P-weighted strategy. Furthermore, B-loc, I-loc, B-time, I-time and O labels were evaluated.

4.3.2. Preprocessing

Lemmatization vs. Stemming

As mentioned in Section 3.2.1, describing the preprocessing applied for the tweets classification model. To evaluate the performance, the results for Lemmatization and Stemming were compared.

Stopwords

Similarly, Section 3.2.1 mentions how cleaning or not the tweet stopwords influences the performance of the model. To compare the results, this aspect was evaluated with the doc2vec, TF-IDF and BERT embedding methods.

4.3.3. Embedding Methods

To evaluate the doc2vec embedding method, explained in Section 3.2.2, the results were compared with other techniques used in the literature on incident reporting in social networks, such as TF-IDF and BERT.

DBOW/doc2vec

For training the doc2vec DBOW model, a dataset was built by pooling the tweets extracted from the collection filters in Section 3.1.2 (50% from the Bogota stream filter and 50% from the other filters). The dataset has in total 1,145,021 tweets from October to December 2018. Using doc2vec for training, tweets collected in real time from Bogota with Stream API allowed capturing non-formal language by users and posts with varied context, not only related to accidents. Finally, the model was trained by splitting the tweet into words, commonly referred to as unigrams or tokens. This and other hyperparameters were selected from a literature review [24,30], including the recommendations of doc2vec [38] authors.

- Splitting the tweet into unigrams.
- Vector size generated in 200 columns.
- Learning rate 0.025 and stochastic gradient descent optimizer.
- Context window size to five—that is, counting two neighboring words to the right and left of the target word.
- Ignoring words when the number of occurrences is fewer than four.
- Twelve iterations and fifteen training epochs.

TF-IDF

For training the model, the necessary parameters were selected for TF-IDF. As previously mentioned, the parameter values were found by performing an optimized search using the grid search method. The parameters taken into account for the search and fit are:

- max_df, to ignore words or tokens with higher frequency.
- max_feautres, to determine the size of the resulting vector.
- min_df, to ignore words with lower frequency.
- ngram_range, minimum and maximum range of N grams for vocabulary construction.

BERT

For this study, the 12-layer model of Cañete et al. [42] was used, which was trained using a Spanish corpus and the case-sensitive version. Unlike the two previous methods, in this case the following preprocessing steps were implemented.

- The @usernames were replaced by the [MASK] label.
- URLs, repeated continuous characters, the RT word and special characters except for punctuation marks were removed. In this case, BERT assigns a special mask.
- Hashtags or # were expanded.

4.3.4. Comparing the Classification Methods

Several experiments were performed using SVM, Naive Bayes, Random Forest and Neural Networks as classifying models, combining the efficiency of each preprocessing and embedding method in Section 3.2.

SVM

As mentioned earlier, the parameters used for TF-IDF and SVM, such as **kernel, gamma** and **C**, were found using the grid search technique.

Naive Bayes

With this model, no modifications were made to the parameters. It was determined whether a tweet was related to an accident or not by simply using the training dataset. The *GaussianNB* class from the *Sklearn* library was used.

Random Forest

For the tests with this model, the RF hyperparameter values were found using the grid search technique of the *Sklearn* library. The parameters selected were the number of trees, the maximum depth of the trees, the minimum number of samples needed to split an internal node and the minimum number of samples needed in a leaf node.

Neural Network

As the NN algorithm is stochastic, to ensure reproducible results, 100 experiments were performed using the dataset of 3804 tweets, splitting this set into 70% tweets for training and 30% for testing. At the end, the average and standard deviation were calculated for the *Accuracy, Recall, Precision* and *F1-score* results obtained in each experiment.

The neural network architecture selected was *Multilayer Perceptron*. It had a hidden layer of *512 units* with a *ReLU* activation function and an output layer with a single neuron

with a *sigmoid* activation function. It was trained with a *batch size* of 32; 25 epochs; an *learning rate* of 1×10^{-4} and the *Adam* optimizer. These hyperparameters were selected by manually adjusting the values until a better result was achieved, because it is a stochastic algorithm, 100 experiments were performed for each fit with the same partitioning of the data for training and testing (see Section 4.2.1), at the end the F1 metric of all experiments was averaged and compared with the results of another fit.

4.3.5. Comparing the Sequence Labeling Models

In order to select an automatic model for entity recognition in Spanish on twitter, the performance of four different state-of-the-art methods was compared: CRF, BiLSTM, BiLSTM+CRF and Spacy.

The experiment employed a training and test set with 1072 and 268 tweets, respectively. Each tweet sentence required the same length of words or tokens for the configuration of the models mentioned above, excluding Spacy library. To define this length, all tweets in the selected corpus were compared and a maximum length of 70 tokens was eventually determined as shown in Figure 4. In this case, for tweets with lower length, a special token was added at the end of the sentence until reaching the desired length. In Figure 4, it can also be seen that most accident related tweets are short in length, both for the train set and test set containing between 15 and 40 tokens.

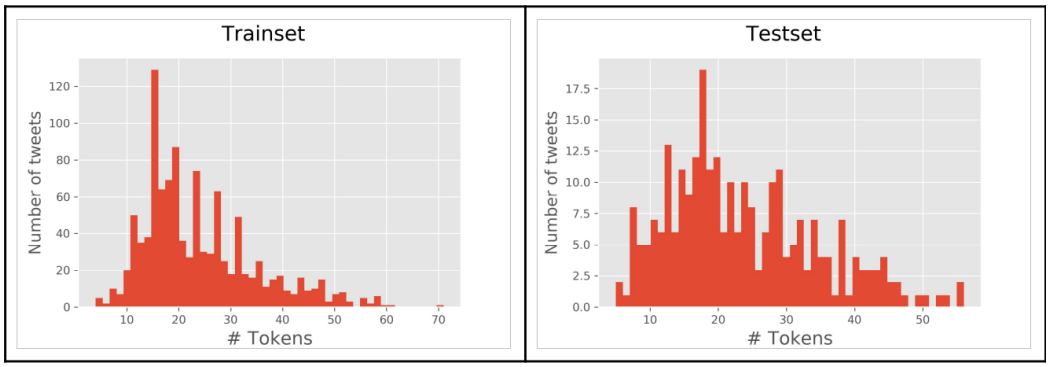

Figure 4. Distribution of number of tokens per number of tweets.

The neural networks models receive a vector representation as input of each token or word. This experiment implemented the *FastText* Spanish model called *cc.es.300.bin* (Available on https://fasttext.cc/docs/en/pretrained-vectors.html, accessed on 20 November 2021). The design of the experiment takes into account that neural networks are stochastic algorithms. Therefore, 25 experiments were performed and then the arithmetic average of each metric was calculated. For each network, it was specified a batch size of 16 and 40 epochs, the latter using the early stop strategy to get the best iteration.

CRF

For feature extraction with tweets, the steps proposed by Taufik et al. [45], Okur et al. [39] and García-Pablos et al. [46] were followed. As a result, 13 functions were defined: (1) word; (2) lowercase; (3) last three letters; (4) last two letters; (5) size or number of characters; (6) pattern or shape function; (7) is Upper function, to determine if the word is uppercase; (8) is Title function, to determine if the first letter is uppercase; (9) is Digit function, to determine if it is a number; (10) POS tag; (11) last two characters of the POS; (12) NER pre-label using CoreNLP; and (13) word embedding of the current word using FastText, which is also ideal for *out-of-vocabulary* processing. Finally, the CRF model was trained using the Python library *sklearn_crfsuite* and setting the algorithm **lbfgs**, and the parameters **c1** and **c2** with 0.1, all transitions = True and **100 iterations**. These parameters were selected according to recommendations found in the related literature [39,45,46].

BiLSTM

The BiLSTM architecture parameters consisted of two LSTM networks, each with *100 units* and *0.5 dropout*. The final layer was *fully connected* with a *softmax* activation function. This layer had 5 units representing each NER label to be predicted as output. Furthermore, finally, the *ADAM* optimizer was used with a learning rate of 0.005, a batch size of 16 and 12 epochs. These architectures and fits of the parameters were initially selected based on literature recommendations [31,32]. The parameters were manually fitted until the final selected configuration was reached with the same strategy mentioned above for the neural networks with 100 experiments and early stopping with patience equal 20. Figure 5 shows an outline of the architecture, taking as input the vector representation of each token using doc2vec and the output the IOB label of each one.

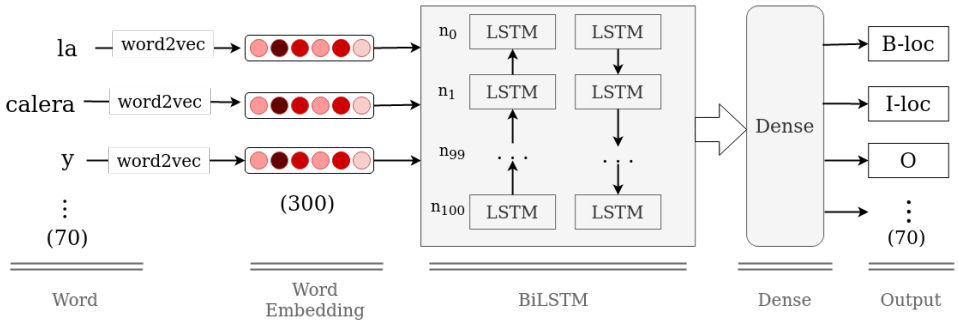

Figure 5. Sequence labeling using BiLSTM architecture.

BiLSTM + CRF

The defined implementation is similar to the previous model; word embedding is performed with FastText; BiLSTM is applied with 100 units and 0.5 dropout. In this case, the fully connected layer used the ReLU activation function; the output layer was a CRF layer with 5 units representing each resulting NER label. The ADAM optimizer was used with a lower learning rate of 0.005, a batch size of 16 and 40 epochs. These architectural and parameter fits were selected following the initial recommendations of Aguilar et al. [32]. The parameters were fit in the same way as the previous BiLSTM architecture mentioned above.

Spacy

This library already has a pre-trained model available that recognizes several entities such as PERSON, ORG, GPE, LOC, DATE, TIME, PERCENT, among others. For our approach, the Spanish model es_core_news_lg was re-trained using our own dataset of tweets, only for the LOC and TIME labels. For re-training, 500 iterations were used as suggested in the library documentation. As input, it received each tweet sequence with its respective label.

4.4. Results

To evaluate our approach and the different machine learning models, the results were clustered according to the TA/NTA classification models and the sequence labeling models for LOC and TIME labels.

4.4.1. Evaluation on Classification Models

As specified in the experiment design, the proposed SVM model was compared with other classification algorithms used in the literature on social networks. In addition, pipelines were created by combining different normalization and embedding processes. Although different pipelines were explored, Table 7 compares the results when applying normalization according to lemmatization (lemma) or stemming (stem), cleaning or not stopwords and using SVM classification model. As shown in Table 7, in general better

results are obtained when stopwords are not eliminated. In this case, it is recommended to use algorithms such as TF-IDF to filter the words with the highest occurrence, avoiding the elimination of prepositions—such as "entre/between" and "con/with"—, which are regularly used to refer to a traffic accident report. In case of normalization, both lemmatization and stemming obtained similar results. So, it is recommended to test both and select the one that best suits the classification problem.

The state-of-the-art embedding methods are neural networks based algorithms. For example, the results of doc2vec in Table 7 outperformed TF-IDF by 0.5% on average, in all metrics. In the case of BERT trained with Cañete et al. [42] dataset, its results did not outperform those of doc2vec and TF-IDF. This is because during training, Cañete et al. [42] dataset did not consider tweets within the corpus. So, it is clear that in order to work with tweets, it is better to train an embedding method with social media posts. Table 7 shows that the best result is our proposed approach, which consists of applying lemmatization and not eliminating stopwords.

Comparing the best performance results of all models in Table 8, the results between SVM and NNs are similar with an average difference of 0.57% in all metrics. Therefore, both algorithms can be considered when working with tweets. However, our proposed approach with SVM obtained a superior performance with **96.8% accuracy** and **F1-score**. The parameters used for SVM are *gamma of 0.2*, *C* equal to 7 and *RBF kernel*. Finally, although Naive Bayes has long been used in the literature for classification tasks [11,25], when working with informal texts in Spanish as in social media, its results were 3% below the others. Therefore, it can be considered as discarded for accident reports processing. In Table 8, only the best tweet processing pipelines were considered for each classification algorithm.

Table 7. Comparison of preprocessing, embedding and SVM model results.

Preprocessing	Embedding	Acc (%)	F1 (%)	R (%)	P (%)
Lemma	**Doc2vec (*our)**	**96.85**	**96.85**	**96.85**	**96.85**
	TF IDF	96.69	96.69	96.69	96.71
Lemma + Stopwords [a]	Doc2vec	96.63	96.63	96.63	96.65
	TF IDF	96.19	96.19	96.20	96.22
Stem	Doc2vec	96.14	96.14	96.14	96.16
	TF IDF	96.69	96.69	96.69	96.71
Stem + Stopwords [a]	Doc2vec	96.84	96.84	96.84	96.86
	TF IDF	96.03	96.03	96.04	96.05
Cleaning steps for BERT	BERT	95.37	95.37	95.37	95.40

[a] Indicates that stopwords have been removed.

Table 8. Comparison of the best results of the classification algorithms.

Pipeline Model	Acc (%)	F1 (%)	R (%)	P (%)
doc2vec + lemma + SVM (*our)	**96.85**	**96.85**	**96.85**	96.85
TF IDF + stem + NB	93.40	93.40	93.40	93.45
TF IDF + lemma + RF	96.08	96.08	96.08	96.11
TF IDF + stem + NN	96.22	96.12	95.17	**97.09**

4.4.2. Evaluation on Named Entity Recognition Models

In order to assess the performance of the four models presented in Section 3.3.2, the test dataset with 268 tweets and label distribution was employed as shown in Table 6. In the results analysis, the "Other" label was not taken into account since the objective was to assess the detection of location and time labels. Table 9 compares the results of the metrics for the CRF, BiLSTM, BiLSTM+CRF and SpaCy models. The best performing model was the Spacy re-trained model, which had an F1 score of 91.97%, followed by CRF with 91.66%.

Using a pre-trained model such as SpaCy is convenient for developers and researchers due to its easy and fast implementation. In addition, the results in Table 9 demonstrate a better performance against the other models. Finally, the detailed results of the re-trained SpaCy model are shown in Table 10. It had better results with the labels that have more tokens in the corpus, such as the location tags.

Table 9. Performance of the proposed models (Best F1 score).

Model	F1 (%)	Recall (%)	Precision (%)
CRF	91.66	89.76	**93.80**
BiLSTM	90.88	89.77	92.25
BiLSTM + CRF	84.22	81.55	88.50
SpaCy	**91.97**	**91.97**	92.17

Table 10. Performance of re-trained SpaCy.

Best Model	# Entities	F1 (%)	Recall (%)	Precision (%)
B-loc	369	88.70	87.82	89.60
I-loc	893	94.83	95.82	93.86
B-time	33	62.22	51.85	77.78
I-time	33	74.51	65.52	86.36
Overall	**1328**	**91.97**	**91.97**	**92.17**

4.4.3. Comparison of Literature Related

Then, our approach was selected and compared with studies related to traffic events detection in social media. The studies were selected for having similarities with the study area and tweets in Spanish. This comparison is intended to give us a clear idea of the performance of our method with respect to other works. It should be clarified that for this comparison each paper mentioned in Tables 11 and 12 was performed with its own dataset and other classification models, besides being oriented to different cities. The results shown in the tables are mentioned by the authors in the results of each paper.

Classification

Our proposed traffic tweet classification model for the city of Bogota obtained an **F1-score** of **96.8%**. It was higher in comparison with similar studies as shown in Table 11. Each study mentioned in this table compares the language used, the target region, the classifier used and the classes to be predicted.

Table 11. Comparison between related studies and the proposed classification model in terms of language, region and F1 score.

Author	Language	Region	Classifier	Class	F1 (%)
Our approach	Spanish	Bogotá, Colombia	Do2vec + SVM	Accident Incident	**96.85**
Arias et al. [12]	Spanish	Cuenca, Ecuador	BoW + SVM	Traffic Incident	85.10
Caimmi et al. [27]	Spanish	Buenos Aires, Argentina	TF IDF + Ensemble SVM, SMO, NB	Traffic Incident	91.44
Pereira et al. [24]	Portuguese	Brasil	BoW + word2vec + SVM	Travel-Related	85.48

Location Recognition

The proposed approach extracted the location and time labels, obtaining an **F1 score** of **91.97%**. It was higher in comparison with similar studies as shown in Table 12. For this comparison we mention the language used, the target region, the sequential labeling model

and the labels extracted for each task. We obtained these results from the corresponding papers cited in the table.

Table 12. Comparison between related studies and the Named Entity Recognition (NER) model in terms of language, region and F1 score.

Author	Language	Region	NER	Classes	F1 (%)
Our approach	Spanish	Bogotá, Colombia	Spacy retrained with tweets	Loc, Time	**91.97**
Arias et al. [12]	Spanish	Cuenca, Ecuador	Rule-Based	Loc	80.61
Gelernter & Zhang [36]	Spanish	Spanish Tweets	Rule-Based + NER Software + Translate	Toponymy	86.10
Sagcan & Karagoz [37]	Turkish	Turkish Tweets	Rule-Based + CRF	Loc	62.00

4.4.4. Analysis of the Processing of Tweets in Bogotá

Once the models of our approach were trained, the pipeline defined in Figure 1 of Section 3 was applied. As mentioned above, 4,973,900 tweets were collected from October 2018 to July 2019. Table 13 shows the number of tweets classified as TA and NTA according to each filter defined in Section 3.1.2.

Table 13. Number of tweets classified as TA/NTA according to collection filter; and number of tweets extracted with location information.

Collection Filter	Non-Accidents		Accidents		Tweets Location
	# [a]	% [b]	#	%	
Stream Bogotá	4,021,481	99.85	5832	0.15	4463
Stream Follow/Timeline User	487,545	84.8	87,271	15.2	80,277
Search Token	210,183	77.5	60,970	22.5	54,765
Search Timeline User	50,507	50.2	50,111	49.8	47,398

[a] Indicates the number of tweets classified by the collection filter. [b] Indicates the percentage of tweets classified by the collection filter.

The amount of tweets filtered by the classifier varies according to the collection filter used. For example, about 0.15% of tweets in the *Stream Bogotá (geotagged)* dataset are selected as traffic accidents, being mostly posts of @BogotaTransito. However, there are about 2000 tweets from other users; some are journalists or newscasts. On the other hand, the *Twitter Search API* tool allowed to extract a greater number of tweets related to accidents. With the *Search Token*, 22.5% of the posts were classified, and with the *Search Timeline User*, 50% of the tweets were classified. The latter dataset was extracted directly from previously selected official users. Note that these sets may have tweets in common; they were filtered in the following phases of location detection.

The trained classification method had an accuracy of 96.8%. By observing the tweets filtered in the automatic classification process, it was evident that the model effectively discriminates between tweets related to accidents and tweets that are related to other types of incidents, such as stranded vehicles, protests, slow traffic, among others. This is because these features were included at the beginning, during the collection and training phases. Table 14 shows some examples where these posts share similar content and are not related to accidents. Furthermore, it shows some examples of accident-related tweets that the classifier correctly predicted. On the other hand, the 3.2% classification error was mitigated in the next phase of named entity recognition, where tweets without location references were discarded.

The entity recognition phase consisted of extracting the locations and coordinates referenced in the accident reports. The datasets were taken using the collection filter, and the location entities were extracted from the tweets. The number of filtered tweets with locations is shown in Table 13. The next step was to cluster the four extracted datasets and eliminate the tweets repeated, resulting in a final dataset with 84,262 accident tweets with recognized entities. As mentioned in the geolocation phase, some of these tweets

refer to the same accident but were reported with a time lag causing noise for the analyses. For instance, the official users of the Secretariat of Mobility (*@BogotaTransito*) and the transportation company Transmilenio (*@rutasitp*) reported the same incident with more than 2 h of difference. For this reason, it was decided to discard the tweets of rutasitp. Likewise, duplicate accident tweets that refer to the same location in a 1-h time window were discarded. Additionally, tweets related to other types of incidents or traffic events that the classifier failed to filter out were removed by matching keywords such as "incidente", "accidente", "choque" (incident, accident and crash in English, respectively) and others keywords relevant to accidents. Consequently, it was possible to eliminate spam tweets from bots such as *@nikolai68843464* and some retweets. In this process, a final set of 43,235 unique tweets was extracted and finally processed through the geocoder tool called Batch Geocode, combining the results from Google Maps, OpenStreetMap and GeoNames. Finally, as a result, the coordinates of **26,362 tweets** were generated. This last set generated was considered the definitive set and was used for the coverage analysis detailed below. Table 15 shows the number of accident-related tweets with locations per month before and after the geocoder processing.

Table 14. Correct accuracy analysis.

Tweet	Label Prediction
Semáforos de la Carrera 24 con Calle 9 en amarillo intermitente, Tanto por la calle como por la carrera con riesgo de incidente vehicular	No accident
Inicia marcha SENA Kra 30A esta hora inicia desplazamiento de estudiantes del SENA sede carrera 30 con calle 14 por toda la Av. NQS hacia el norte, utilizando calzada mixta con afectación de calzada de TransMilenio.	No accident
Hueco causa accidentalidad Cra. 68c #10-16 sur, Bogotá	Accident
A esta hora nuestras unidades brindan apoyo en la Av primero de mayo por 24, donde se presenta un choque entre un vehículo particular y una motocicleta.	Accident

Table 15. Number of tweets related to traffic accidents with location and coordinates per month.

Month	# Tweets TA	# Tweets Coordinates
October	3682	2194
November	4072	2358
December	3634	2114
January	4316	2692
February	4127	2534
Marh	4029	2500
April	3697	2241
May	4123	2545
June	5281	3287
July	6274	3897
Total	43,235	26,362

4.5. Analysis of Traffic Accident Coverage between Twitter and Official Source

Coordinates obtained by the geoparsing method were used to perform a comparison of accident patterns between Twitter and the official source. The two sets are from October 2018 to July 2019. The official accident information had **25,299** records and the set of tweets had **26,362** records, respectively; both included the latitude/longitude coordinates and date/time of the incident.

4.5.1. Map-Matched with Official Accident Record

Traffic accidents reported on Twitter may contain additional information. To address this issue, the percentage of accidents from official records that are covered by Twitter was quantified. Because accidents are not reported immediately, a time and distance radius must be established to match reports between Twitter and the official source. Zhang et al. [13] and Gu et al. [11] proposed to consider aspects specific to the study area such as size, traffic congestion and emergency response time. Taking into account the above, in order to match the records between Twitter and the official source, a **radius of 1 km distance and 2 h difference was defined**. Bogota was the city with the highest traffic congestion in the world in 2019 (According to the 2019 INRIX Global Traffic Scorecard), delaying the response to accidents and impeding to determine the exact time they occurred, with 1 or 2 h difference. In addition, the locations described by users on Twitter are sometimes inaccurate and have a margin of error. They are also affected by geocoder's accuracy. For instance, in tests performed on 1% of the tweets, it only obtained 78.7% correct coordinates—those with less than 1 km difference between prediction and real value.

Table 16 shows the results of geopairing using 1 km and 2 h difference. Out of the 26,362 tweets, 8619 (32.7%) of these were matched with the official source, with an average distance of 436 m and 47 min difference. In addition, 2896 (34%) of the tweets were posted before the official record and 5723 (66%) were posted later, respectively. Some examples of these tweets are shown in Table 17 and their time difference information before or after with the official matched report, in some cases the time difference is more than one hour. An additional sample of 1431 tweets that were posted by individual users (excluding BogotaTransito) was taken. Following the recommendations of Gu et al. [11] to measure the proportion of additional information, 455 posts (31.8%) were matched with an average difference of 460 m and 51 min. In this case, 164 (36%) were posted before the official record and 291 (64%) later, respectively.

The unmatched tweets can be considered as additional reports and correspond to 17,743 tweets, of which 976 are from individual users, such as support networks (@RedapBogota) and reporters. Manually examining some records and with the support of Tables 17 and 18, the following was discussed:

- In some cases, there is a difference of up to 1 h before or after between the official source and Twitter. The official hour is conditioned by the witnesses' perception of time, while Twitter may be conditioned by the duration of time it takes for users to arrive at the scene. These limitations restrict us to know the exact time of the incidents.
- There are differences of more than 1 km between the actual and predicted coordinate. One of the reasons is the inaccuracy of the accident address described on Twitter and the geocode used in Bogota.
- Not all incident-related tweets from @BogotaTransito are posted on the official source. These tweets can be considered as additional reliable information since they are posts from an official transit user, and therefore can be included within the tweets of individual users.

Table 16. Number of accidents reported according to data source.

Data Source	**Number of Reports**	
Official data	25,299	
	All reports	**Excluding @BogotaTransito**
Twitter data	26,362	1431
Twitter (matched by Official data in 1 km and 2 h)	8619	455
# Twitter "additional" reports	17,743	976

Table 17. Tweets map-matched by accident official data.

Tweet [a]	Time Difference
Calle 55 sur carrera 19B Choque con herido ya hay Ambulancia se necesita @TransitoPolicia @BogotaTransito @SectorMovilidad @TransitoBta	23 s later
Aparatoso accidente en la Av. Córdoba con calle 127 sentido sur-norte. Trancón, se recomienda usar vías alternas. @gusgomez1701 @CaracolRadio @SectorMovilidad	58 min later
Incidente vial entre dos particulares en la Calle 19 con Carrera 34, sentido occidente-oriente. Unidad de @TransitoBta asignada.	1 h 37 min later by BogotaTransito
@BogotaTransito buenos días, necesitamos su ayuda en la carrera 7 con calle 163, choque de sitp con taxi.	2 min earlier
@TransMilenio @PoliciaBogota Peatón atropellado en troncal calle 80, estación Av 68, Se requiere ambulancia urgente!!!!!	15 min earlier

[a] Some special characters and emoticons were removed.

Table 18. Tweets not map-matched by accident official data.

Tweet [a]	Issue
@TransitoBta accidente en sentido S-en la 27 sur con 10 monumental trancon @Citytv @NoticiasCaracol @PoliciaColombia	Vague location description
Incidente vial entre particular y un ciclista en la Calle 65A con Carrera 112, sentido occidente-oriente. Unidad de @TransitoBta y asignadas.	Prediction of coordinates > 1 km difference; tweet posted by BogotaTransito
Accidente de 2 vehículos calle 161 con carrera 7, sin heridos solo latas, generan afectación del tráfico.	No map-matched
Incidente vial entre particular y ciclista, en la Autonorte con calle 170, sentido sur-norte.Unidad de @TransitoBta y asignada.	No map-matched; tweet posted by BogotaTransito

[a] Some special characters and emoticons were removed.

4.5.2. Analysis of the Accident Pattern in Time and Space

Then, the accidents on Twitter and official records are examined over time. Figure 6 shows the percentage of reports by official sources, @BogotaTransito tweets and individual users by time of day. Accident reports match with the peak traffic hours, between 5 a.m. and 8 p.m. These hours coincide with the working day in Colombia. In fact, the three sources coincide with the peak hours that correspond to the hours when workers spend in traffic—around 7 a.m. and 6 p.m. For the other hours of the day the activity on social networks decreases. Surprisingly, the reporting activity between @BogotaTransito's tweets and the official data is not consistent in the evening hours, even though the data is reported by the same official entity. In this case, according to Gu et al. [11], an advantage of accident detection on Twitter is the higher coverage during daytime. However, it is not as effective at night and in the early morning as is the case with official data. As for the number of accident reports by day of the week in Figure 7, the data also varies depending on the usual workday, with higher activity on weekdays and fewer tweets on weekends. This can be seen for both the official and Twitter datasets.

Finally, a comparison of the spatial pattern of accidents between both sources was performed with a heat map, but using the *Kernel Density Estimation* (KDE) method for accident intensity per square meter. In the analysis to calculate the bandwidth of the KDE method, the Scott Factor and the Gaussian Kernel were used.

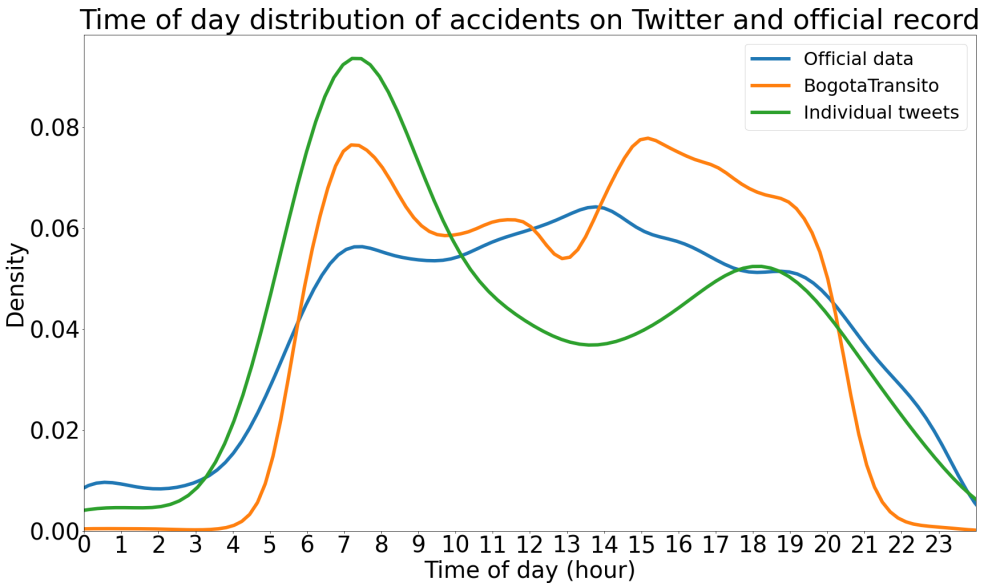

Figure 6. Distribution of accidents reported on Twitter and the Official Register by time of day.

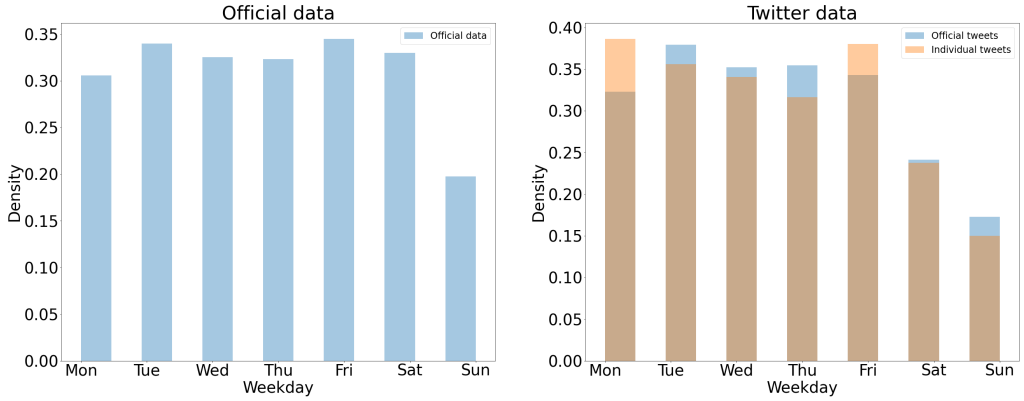

Figure 7. Distribution of reported accidents on Twitter and Official Register by day of the week.

Figure 8 shows the distribution of accidents from October 2018 to July 2019, according to Twitter and official data. Accidents on Twitter are distributed spatially similar to the records in the official data source, maintaining a higher accident rate in the center of Bogota and decreasing gradually as it moves away from the center. Some regions match the concentration of accidents, such as the localities of Chapinero and the limits between Martires and Puente Aranda. These areas are characterized by being the main trade center and the industrial zone of Bogota with the highest traffic activity. As for the western region, both data sources almost match accident areas in the localities of Engativa, Fontibon and Kenedy. However, Twitter posting activity is somewhat lower in these areas. In conclusion, most of the Twitter data and the official source spatially match the density of accidents. Regarding the Twitter extraction process, it performed better in the regions with more commercial and industrial activity where Twitter users are more frequent, while in other areas Twitter publications decreased.

Finally, the results of the above comparisons—in terms of time and space—provide greater confidence and credibility in the effectiveness of the accidents reported on Twitter as additional information. In this sense, according to Zhang et al. [13], these data can be used as complementary sources and not substitutes for existing detection methods; that was validated in the context of Bogota.

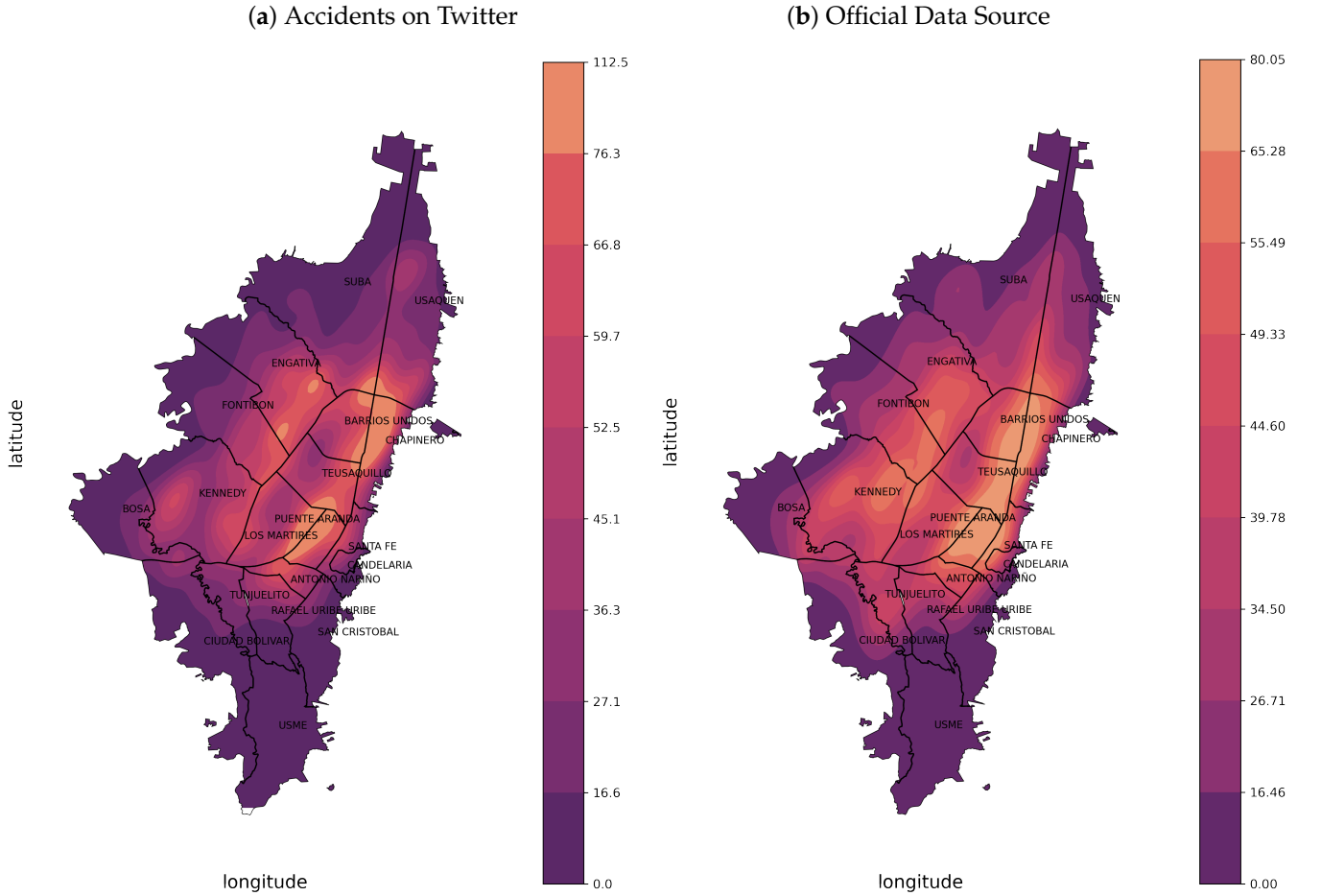

Figure 8. Geographical incidence of accidents reported from October 2018 to July 2019 according to data on Twitter (**a**) and Official Source (**b**).

5. Discussion and Conclusions

This paper presented a method for the extraction of traffic accidents in social networks. It only used Twitter, but other social networks can be implemented. The methodology was divided into four phases: first the collection of tweets, second the classification of accidents, third the detection of locations from the tweet content, and finally the generation of coordinates. The accident tweets were also validated by comparing them with official data from the Bogota Mobility Secretariat.

For tweet classification, different word embedding and classification methods were combined to assess the efficiency of our model to discriminate between traffic accidents and other types of incidents. In order to do so, those types of posts were included while building the dataset collaboratively. Our implementation of *doc2vec* and *SVM* in the classifier achieved an accuracy of 96.8% comparable to the state-of-the-art. It is recommended not to remove stopwords in preprocessing, since word embedding methods work best if most of the tweet content is preserved.

During location extraction from text, a few architectures for named entity recognition were compared. However, our *SpaCy* model re-trained with Spanish tweets achieved superior performance with 91.9% on the F1-score metric. A key contribution of this work is the geolocation phase. Since these are social network posts, the use of informal language and abbreviations must be corrected. To do so, the addresses were normalized as a preliminary step to improve the performance of the geocoders. Hashtags can also include location information [35]. Hence, word segmentation was normalized. Repeated reports and some retweets were also eliminated, since they are often posted with 5 h of difference, generating false alarms.

A benefit of extracting traffic accident reports from Twitter is the possibility to detect additional accidents. To address that issue, data from Twitter and the official Bogota transit source were matched. It was found that about 33% of the tweets were reported by the official source. The tweets that were not matched are additional reports and most correspond to official users such as transit authorities, emergency support networks and reporters. These tweets are considered reliable as they are posted by official accounts. Manually examining the accidents, it was determined that in some cases there is 1 or 2 h difference before or after between the official source report and Twitter, these differences restrict us to define the exact time of the incidents.

The timing and spacing of accident patterns between tweets and official records were also compared. In this case, according to Gu et al. [11], an advantage of accident detection on Twitter is the higher coverage during daytime. However, it is not as effective at night and in the early morning as is the case with official data.

Moreover, most of the Twitter data and the official source spatially match the spatial distribution of accidents. Regarding the Twitter extraction process, it performed better in the regions with more commercial and industrial activity where the users of this social network transit more frequently. Finally, the results provide greater confidence and credibility in the effectiveness of the accidents reported on Twitter as additional information. In this sense, according to Zhang et al. [13], these data can be used as complementary sources and not substitutes for existing detection methods; that was validated in the context of Bogota.

Several opportunities were identified in terms of geolocation that open the possibility for further research. These include resolving inaccuracies or vague references to the accident location on Twitter and correcting common spelling errors. The geocoding tools such as *Google Maps* and *OpenStreetMap* are not sufficient for cities outside the United States. Currently, these tools may have differences of more than 1 km between the actual and predicted coordinate. In further research, this type of error could be analyzed in depth and a specific geocoding for the city being studied could be designed with a better performance. Another factor that distorts the prediction of geocoders are tweets that report more than two accidents with different addresses. For this case, a special treatment should be established for this type of posts. Other advances could be the development of real-time accident monitoring systems for the city of Bogota that include data from social networks. Further research should focus on integrating or merging tweets with other sources for predicting traffic accidents in the city.

Author Contributions: Conceptualization, N.S.-R., C.G.-O. and C.P.; methodology, N.S.-R.; software, N.S.-R.; validation, N.S.-R. and C.G.-O.; formal analysis, N.S.-R.; investigation, N.S.-R.; original draft preparation, N.S.-R.; writing, review and editing, N.S.-R., C.G.-O. and C.P.; visualization, N.S.-R.; supervision, C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data sets for the training of the different models are available on:—Nestor Suat-Rojas, Camilo Gutierrez-Osorio, & Cesar Pedraza Bonilla. (2021). Dataset of accidents reported on Twitter Colombia (1.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.5548475, accessed on 20 November 2021—Gutierrez-Osorio, Camilo; Suat, Nestor; Pedraza, Cesar (2020), "Bogota city traffic accidents—Social media datasets", Mendeley Data, V1, https://doi.org/10.17632/c2r6tk9hbg.1, accessed on 20 November 2021.

Acknowledgments: The authors thank the National University of Colombia and the University of Los Llanos for opening an agreement for the development of research. Furthermore, a thank you to the Programming Languages and Systems (PLaS) research group for their feedback on the work.

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

References

1. Cookson, G.; Pishue, B. *INRIX Global Traffic Scorecard*; Number February; INRIX Research: Kirkland, WA, USA, 2018; p. 44.
2. Víctimas Fallecidas y Lesionadas Valoradas por INMLCF. Nacionales. Agencia Nacional de Seguridad Vial. 2017. Available online: http://ansv.gov.co/observatorio/?op=Contenidos&sec=63&page=20 (accessed on 20 November 2021).
3. Wang, S.; He, L.; Stenneth, L.; Yu, P.S.; Li, Z. Citywide traffic congestion estimation with social media. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems—GIS '15, Bellevue, WA, USA, 3–6 November 2015; pp. 1–10. [CrossRef]
4. Fan, X.; He, B.; Wang, C.; Li, J.; Cheng, M.; Huang, H.; Liu, X. Big Data Analytics and Visualization with Spatio-Temporal Correlations for Traffic Accidents. In Proceedings of the 15th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2015), Zhangjiajie, China, 18–20 November 2015; Volume 9531, pp. 255–268. [CrossRef]
5. Subaweh, M.B.; Wibowo, E.P. Implementation of Pixel Based Adaptive Segmenter method for tracking and counting vehicles in visual surveillance. In Proceedings of the 2016 International Conference on Informatics and Computing (ICIC 2016), Mataram, Indonesia, 28–29 October 2016; pp. 1–5. [CrossRef]
6. Li, L.; Zhang, J.; Zheng, Y.; Ran, B. Real-Time Traffic Incident Detection with Classification Methods. In *Green Intelligent Transportation Systems, Lecture Notes in Electrical Engineering*; Springer: Singapore, 2018; Volume 419, pp. 777–788. [CrossRef]
7. Zhang, S.; Wu, G.; Costeira, J.P.; Moura, J.M. FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3687–3696. [CrossRef]
8. Krausz, N.; Lovas, T.; Barsi, Á. Radio frequency identification in supporting traffic safety. *Period. Polytech. Civ. Eng.* **2017**, *61*, 727–731. [CrossRef]
9. Wang, S.; Zhang, X.; Cao, J.; He, L.; Stenneth, L.; Yu, P.S.; Li, Z.; Huang, Z. Computing Urban Traffic Congestions by Incorporating Sparse GPS Probe Data and Social Media Data. *ACM Trans. Inf. Syst.* **2017**, *35*, 1–30. [CrossRef]
10. Kuflik, T.; Minkov, E.; Nocera, S.; Grant-Muller, S.; Gal-Tzur, A.; Shoor, I. Automating a framework to extract and analyse transport related social media content: The potential and the challenges. *Transp. Res. Part C Emerg. Technol.* **2017**, *77*, 275–291. [CrossRef]
11. Gu, Y.; Qian, Z.S.; Chen, F. From Twitter to detector: Real-time traffic incident detection using social media data. *Transp. Res. Part C Emerg. Technol.* **2016**, *67*, 321–342. [CrossRef]
12. Arias, B.; Orellana, G.; Orellana, M.; Acosta, M.I. *A Text Mining Approach to Discover Real-Time Transit Events from Twitter*; Springer International Publishing: Cham, Switzerland, 2019; Volume 884, pp. 266–280. [CrossRef]
13. Zhang, Z.; He, Q.; Gao, J.; Ni, M. A deep learning approach for detecting traffic accidents from social media data. *Transp. Res. Part C Emerg. Technol.* **2018**, *86*, 580–596. [CrossRef]
14. Sherif, H.M.; Shedid, M.; Senbel, S.A. Real Time Traffic Accident Detection System using Wireless Sensor Network. In Proceedings of the 2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR), Tunis, Tunisia, 11–14 August 2014; pp. 59–64. [CrossRef]
15. Aslam, J.; Lim, S.; Pan, X.; Rus, D. City-scale traffic estimation from a roving sensor network. In *SenSys 2012—Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems*; Association for Computing Machinery: New York, NY, USA, 2012; pp. 141–154. [CrossRef]
16. Zuo, W.; Guo, C.; Liu, J.; Peng, X.; Yang, M. A police and insurance joint management system based on high precision BDS/GPS positioning. *Sensors* **2018**, *18*, 169. [CrossRef] [PubMed]
17. Kwak, H.c.; Kho, S. Predicting crash risk and identifying crash precursors on Korean expressways using loop detector data. *Accid. Anal. Prev.* **2016**, *88*, 9–19. [CrossRef] [PubMed]
18. Li, Z.; Cha, S.K.; Wan, C.; Cui, B.; Zhang, N.; Xu, J. Detecting Anomaly in Traffic Flow from Road Similarity Analysis. In *17th International Conference, WAIM 2016, Proceedings, Part II*; Cui, B., Zhang, N., Xu, J., Lian, X., Liu, D., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9659, pp. V–VI. [CrossRef]
19. Chen, Q.; Song, X.; Yamada, H.; Shibasaki, R. Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident Inference. In Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16), Phoenix, AZ, USA, 12–17 February 2016; pp. 338–344.
20. Petalas, Y.G.; Ammari, A.; Georgakis, P.; Nwagboso, C. *A Big Data Architecture for Traffic Forecasting Using Multi-Source Information*; ALGOCLOUD 2016; Springer International Publishing: Cham, Switzerland, 2017; Volume 10230, pp. 65–83. [CrossRef]
21. Salas, A.; Georgakis, P.; Nwagboso, C.; Ammari, A.; Petalas, I. Traffic Event Detection Framework Using Social Media. In Proceedings of the IEEE International Conference on Smart Grid and Smart Cities, Singapore, 23–26 July 2017; p. 5. [CrossRef]
22. Salas, A.; Georgakis, P.; Petalas, Y. Incident Detection Using Data from Social Media. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 751–755. [CrossRef]

23. Kurniawan, D.A.; Wibirama, S.; Setiawan, N.A. Real-time traffic classification with Twitter data mining. In Proceedings of the 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 5–6 October 2016; pp. 1–5. [CrossRef]
24. Pereira, J.; Pasquali, A.; Saleiro, P.; Rossetti, R. Transportation in Social Media: An Automatic Classifier for Travel-Related Tweets. In Proceedings of the 18th EPIA Conference on Artificial Intelligence (EPIA 2017), Porto, Portugal, 5–8 September 2017; Volume 8154, pp. 355–366. [CrossRef]
25. Nguyen, H.; Liu, W.; Rivera, P.; Chen, F. TrafficWatch: Real-Time Traffic Incident Detection and Monitoring Using Social Media Hoang. In *PAKDD: Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Bailey, J., Khan, L., Washio, T., Dobbie, G., Huang, J.Z., Wang, R., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9651, pp. 540–551. [CrossRef]
26. Schulz, A.; Ristoski, P.; Paulheim, H. I see a car crash: Real-time detection of small scale incidents in microblogs. In *The Semantic Web: ESWC 2013 Satellite Events. ESWC 2013. Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 22–33. [CrossRef]
27. Caimmi, B.; Vallejos, S.; Berdun, L.; Soria, Ï.; Amandi, A.; Campo, M. Detección de incidentes de tránsito en Twitter. In Proceedings of the 2016 IEEE Biennial Congress of Argentina (ARGENCON 2016), Buenos Aires, Argentina, 15–17 June 2016; pp. 1–6. [CrossRef]
28. Gutiérrez, C.; Figueiras, P.; Oliveira, P.; Costa, R.; Jardim-goncalves, R. An Approach for Detecting Traffic Events Using Social Media. In *Emerging Trends and Advanced Technologies for Computational Intelligence*; Springer: Cham, Switzerland, 2016; Volume 647. [CrossRef]
29. Anantharam, P.; Barnaghi, P.; Thirunarayan, K.; Sheth, A. Extracting City Traffic Events from Social Streams. *ACM Trans. Intell. Syst. Technol.* **2015**, *6*, 1–27. [CrossRef]
30. Chen, Y.; Lv, Y.; Wang, X.; Wang, F.Y. A convolutional neural network for traffic information sensing from social media text. In Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6. [CrossRef]
31. Peres, R.; Esteves, D.; Maheshwari, G. Bidirectional LSTM with a Context Input Window for Named Entity Recognition in Tweets. In Proceedings of the K-CAP 2017: Knowledge Capture Conference (K-CAP 2017), New York, NY, USA, 4–6 December 2017; pp. 1–4. [CrossRef]
32. Aguilar, G.; López Monroy, A.P.; González, F.; Solorio, T. Modeling Noisiness to Recognize Named Entities using Multitask Neural Networks on Social Media. In Proceedings of the NAACL-HLT 2018 Association for Computational Linguistics, New Orleans, LA, USA, 1–6 June 2018; Volume 1, pp. 1401–1412. [CrossRef]
33. Gelernter, J.; Balaji, S. An algorithm for local geoparsing of microtext. *GeoInformatica* **2013**, *17*, 635–667. [CrossRef]
34. Ritter, A.; Clark, S.; Mausam; Etzioni, O. Named entity recognition in tweets: An experimental study. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 27–32 July 2011; pp. 1524–1534.
35. Malmasi, S.; Dras, M. Location Mention Detection in Tweets and Microblogs. In *PACLING 2015, CCIS*; Oxford University Press: Oxford, UK, 2016; pp. 123–134. [CrossRef]
36. Gelernter, J.; Zhang, W. Cross-lingual geo-parsing for non-structured data. In Proceedings of the 7th Workshop on Geographic Information Retrieval, Association for Computing Machinery, New York, NY, USA, 5 November 2013; pp. 64–71. [CrossRef]
37. Sagcan, M.; Karagoz, P. Toponym Recognition in Social Media for Estimating the Location of Events. In Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW), Atlantic City, NJ, USA, 14–17 November 2015; pp. 33–39. [CrossRef]
38. Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; Volume 32.
39. Okur, E.; Demir, H.; Özgür, A. Named entity recognition on twitter for Turkish using semi-supervised learning with word embeddings. In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), Portoroz, Slovenia, 23–28 May 2016; pp. 549–555.
40. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. In Proceedings of the 1st International Conference on Learning Representations, ICLR 2013—Workshop Track Proceedings, Scottsdale, AZ, USA, 2–4 May 2013; pp. 1–12.
41. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019.
42. Cañete, J.; Chaperon, G.; Fuentes, R.; Pérez, J. Spanish Pre-Trained BERT Model and Evaluation Data. In Proceedings of the PML4DC at ICLR 2020, Addis Ababa, Ethiopia, 26 April 2020; pp. 1–10.
43. Norvig, P. Natural Language Corpus Data. In *Beautiful Data: The Stories Behind Elegant Data Solutions*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2009; pp. 219–242. ISBN 978-0596157111.
44. Honnibal, M.; Johnson, M. An improved non-monotonic transition system for dependency parsing. In Proceedings of the Conference Proceedings—EMNLP 2015: Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1373–1378. [CrossRef]

45. Taufik, N.; Wicaksono, A.F.; Adriani, M. Named entity recognition on Indonesian microblog messages. In Proceedings of the 2016 International Conference on Asian Language Processing (IALP 2016), Tainan, Taiwan, 21–23 November 2016; pp. 358–361. [CrossRef]
46. García-Pablos, A.; Perez, N.; Cuadros, M. Sensitive Data Detection and Classification in Spanish Clinical Text: Experiments with BERT. In Proceedings of the 12th Edition of Language Resources and Evaluation Conference (LREC2020), Marseille, France, 11–16 May 2019. Available online: http://www.lrec-conf.org/proceedings/lrec2020/pdf/2020.lrec-1.552.pdf (accessed on 20 November 2021).